

# Sequential Bayesian inference for mixture models and the coalescent using sequential Monte Carlo samplers with transformations

Richard G. Everitt, Richard Culliford, Felipe Medina-Aguayo and Daniel J. Wilson

21st December 2016

## Abstract

This paper introduces methodology for performing Bayesian inference sequentially on a sequence of posteriors on spaces of different dimensions. We show how this may be achieved through the use of sequential Monte Carlo (SMC) samplers (Del Moral *et al.*, 2006, 2007), making use of the full flexibility of this framework in order that the method is computationally efficient. In particular we introduce the innovation of using a sequence of distributions that are defined on spaces between which bijective transformations exist, using these transformations to move particles effectively between one target distribution and the next. This approach, combined with adaptive methods and the use of multiple routes between targets, yields an extremely flexible and general algorithm for tackling the aforementioned situation. We demonstrate this approach on the well-studied problem of model comparison for mixture models, and for the novel application of inferring coalescent trees sequentially, as data arrives.

## 1 Introduction

### 1.1 Sequential inference

Much of the methodology for Bayesian computation is designed with the aim of approximating a posterior  $\pi(\theta)$ , or approximating statistics of this distribution. The most prominent approach is to use Markov chain Monte Carlo (MCMC), in which a Markov chain that has  $\pi$  as its limiting distribution is simulated. It is well known: that this process may be computationally expensive; that it may be difficult to tune the method automatically (Andrieu and Thoms, 2008); and that it can be challenging to determine an appropriate burn in for the chain (Brooks and Gelman, 1998) and to determine how long to run the chain for after burn in (Flegal and Jones, 2008). Therefore, designing and running an MCMC algorithm to sample from a particular target  $\pi$  may require much human input and computer time. This creates particular problems if a user is in fact interested in number of target distributions  $(\pi_t)_{t=1}^T$ : using MCMC on each target requires additional computer time to run the separate algorithms and each may require human input to design the algorithm, determine the burn in, etc. This paper has as its subject the task of using a Monte Carlo method to simulate from each of the targets  $\pi_t$  that avoids these disadvantages.

This work is by no means the first to address this situation. Most prominent is the methodology of particle filtering (Gordon *et al.*, 1993) and its generalisation, the SMC sampler (Del Moral *et al.*, 2006), which is designed to tackle exactly this problem. Roughly speaking, the idea of these approaches is to begin by using importance sampling (IS) to find a set of weighted *particles* that give an empirical approximation to  $\pi_0$  then to, for  $t = 0, \dots, T-1$ , update the set of particles approximating  $\pi_t$  such that they, after changing their positions using a kernel  $K_{t+1}$  and updating their weights, approximate  $\pi_{t+1}$ . This approach is particularly useful where neighbouring target distributions in the sequence are similar to each other, and in this case has the following advantages over running  $T$  separate MCMC algorithms.

- The similarity of neighbouring targets can be exploited since particles approximating  $\pi_t$  may not need much adjustment to provide a good approximation to  $\pi_{t+1}$ . We have the desirable property that we find approximations to each of the targets in the sequence. Further, we also may gain when compared to running a single MCMC algorithm to target  $\pi_T$ , since it may be complicated to set up an MCMC that simulates well from  $\pi_T$  without using a sequence of simpler distributions to guide particles into the appropriate regions of the space.
- If the  $\pi_t$  are unnormalised, SMC samplers also provide an unbiased estimates of the normalising constants of the  $\pi_t$ . In the context of each  $\pi_t$  being an unnormalised Bayesian posterior distribution, is the *marginal likelihood* or *evidence*, a key quantity in Bayesian model comparison. We use  $\pi_t$  to represent an unnormalised target throughout the paper, and use  $\bar{\pi}_t$  for its normalised form.

## 1.2 Targets on spaces of different dimensions

In this paper we consider the case where each  $\pi_t$  is defined on a space of different dimension, often of increasing dimension with  $t$ . A standard particle filter is designed to be used in a special case of this situation: the case where  $\pi_t$  is the path distribution in a state space model,  $\pi_t(\theta_{1:t} | y_{1:t})$ . A particle filter exploits the relationship  $\pi_{t+1}(\theta_{1:t+1} | y_{1:t+1}) \propto \pi_t(\theta_{1:t} | y_{1:t}) p(\theta_{t+1} | \theta_t) f(y_{t+1} | \theta_{t+1})$  (where  $p$  is a distribution known as the dynamic model and  $f$  is known as a measurement model) in order to update a particle approximation of  $\pi_t(\theta_{1:t} | y_{1:t})$  to an approximation of  $\pi_{t+1}(\theta_{1:t+1} | y_{1:t+1})$ . In this paper we consider targets in which there is not such a straightforward relationship between  $\pi_t$  and  $\pi_{t+1}$ . There are two widely encountered situations where this is the case, which we summarise following a description of the main contributions of the paper.

### 1.2.1 Contributions

The methodological approach used in this paper is a special case of the SMC sampler outlined in Del Moral *et al.* (2007). We introduce the following innovations within this framework.

- We use a sequence of distributions that are defined on spaces between which bijective transformations exist, and show how to use these transformations to design efficient samplers by reducing the distance between successive distributions.
- We describe the application of this approach to two new applications: sequential inference under the coalescent and Bayesian model comparison.
- We outline how to use the full flexibility of our SMC framework in order that our samplers are efficient, including the use of adaptive methods and a method that allows mixtures of transition kernels.

### 1.2.2 Bayesian model comparison

We consider the problem of model comparison, where each  $\pi_t$  is a different model and there are  $T$  models that can be ordered, usually in order of their complexity. Current techniques for Bayesian model comparison either: move around the joint space of model and parameters (e.g. reversible jump MCMC (RJMCMC) (Green, 1995)); estimate the marginal likelihood of each model separately (e.g. Didelot *et al.* (2011)); or estimate Bayes' factors between two models (e.g. Gelman (1998)). In this paper we propose an approach that visits each model in the order of increasing complexity, estimating Bayes' factors between neighbouring pair of models. In some applications this is advantageous compared to current approaches; we discuss this more thoroughly in section 2.3.

### 1.2.3 Sequential inference of trees

We consider sequential inference where the posterior changes globally on the receipt of each measurement. One case in which this occurs is in the online inference of phylogenetic trees. The example on which we focus in this paper is that of online inference under the coalescent model in population genetics (Kingman, 1982); we consider the case in which we wish to infer the *clonal ancestry* (or *ancestral tree*) of a bacterial population from DNA sequence data. Current approaches in this area use MCMC, which is a limitation in situations where DNA sequence data does not arrive as a batch, such as may happen when studying the spread of an infectious disease as the outbreak is progressing (Didelot *et al.*, 2014). We instead introduce an SMC approach to online inference, inferring posterior distribution as sequences become available (this approach is similar to that of Dinh *et al.* (2016) which was devised simultaneously to our approach). We further envisage that this approach will be useful in cases in which data is available as a batch, through exploiting the well known property that a tree estimated from  $t$  sequences is usually similar to a tree estimated from  $t + 1$  sequences. Thus, although exploring the space of trees for a large number of sequences appears challenging due to the large number of possible trees, through adding leaves one by one the SMC approach follows a path through tree space in which transitions from distribution  $\pi_t$  to  $\pi_{t+1}$  are not challenging. Further, our approach yields more stable estimates of the marginal likelihood of models than current approaches used routinely in population genetics (Drummond and Rambaut, 2007; Xie *et al.*, 2011). This application is discussed further in section 3.

### 1.2.4 Outline of paper

Section 2 describes the methodological approach used in the paper, considering both practical and theoretical aspects, and provides comparison to existing methods. In section 3 we use our methodology for online inference

under the coalescent, using the flexibility of our proposed approach to describe a method for moving between trees. We then provide an example of the use of the methodology for Bayesian model comparison in section 4, examining the well-studied case of inferring the number of components in a Gaussian mixture model. In this section we demonstrate the full flexibility of our proposed approach. In section 5 we review the approach and outline possible extensions.

## 2 SMC samplers with transformations

### 2.1 SMC samplers with increasing dimension

The use of SMC samplers on a sequence of targets of increasing dimension has been described previously, most recently in Everitt *et al.* (2016) and Dinh *et al.* (2016). These papers introduce an additional proposal distribution for the variables that are introduced at each step. In this section we show that this approach can be seen to be a special case of the SMC sampler in Del Moral *et al.* (2007). We note that it is also straightforward to see that this approach is valid through other arguments; the framework in this section is introduced mainly for the purpose of extending it in the following section.

#### 2.1.1 SMC samplers with MCMC moves

To introduce notation, we first consider the standard case in which the dimension is fixed across all iterations of the SMC. For simplicity we consider only SMC samplers with MCMC moves, and we consider an SMC sampler that has  $T$  iterations. Let  $\pi_t$  be our target distribution of interest at iteration  $t$ , this being the distribution of the random vector  $\theta_t$  on space  $E$ . Throughout the paper our notation is that the values taken by particles in the SMC sampler have a  $(p)$  superscript to distinguish them from random variables/vectors; so for example  $\theta_t^{(p)}$  is the value taken by the  $p$ th particle. We define  $\pi_0$  to be a distribution from which we can simulate directly, simulate each particle  $\theta_0^{(p)} \sim \pi_0$  and set its normalised weight  $w_0^{(p)} = 1/P$ . Then for  $0 \leq t < T$  at the  $(t+1)$ th iteration of the SMC sampler, the following steps are performed.

1. **Reweight:** Calculate the updated (unnormalised) weight  $\tilde{w}_{t+1}^{(p)}$  of the  $p$ th particle  $\theta_t^{(p)}$

$$\tilde{w}_{t+1}^{(p)} = w_t^{(p)} \frac{\pi_{t+1}(\theta_t^{(p)})}{\pi_t(\theta_t^{(p)})}. \quad (1)$$

2. **Resample:** Normalise the weights to obtain normalised weights  $w_{t+1}^{(p)}$  and calculate the *effective sample size* (ESS) (Kong *et al.*, 1994)

$$\text{ESS} = \left( \sum_{p=1}^P \left( w_{t+1}^{(p)} \right)^2 \right)^{-1}. \quad (2)$$

If the ESS falls below some threshold, e.g.  $\alpha P$  where  $0 < \alpha < 1$ , then resample. Throughout this paper we use stratified resampling, which is a simple low variance scheme (Doucet and Johansen, 2009).

3. **Move:** For each particle perform an MCMC move with target  $\pi_{t+1}$  to move  $\theta_t^{(p)}$  to  $\theta_{t+1}^{(p)}$ .

This algorithm yields an empirical approximation of  $\pi_t$

$$\hat{\pi}_t^P = \sum_{p=1}^P w_t^{(p)} \delta_{\theta_t^{(p)}}, \quad (3)$$

where  $\delta_\theta$  is a Dirac mass at  $\theta$ , and an estimate of its normalising constant

$$\hat{Z}_t = \frac{1}{P} \prod_{s=1}^t \sum_{p=1}^P \tilde{w}_s^{(p)}. \quad (4)$$

We note that

- In some circumstances it is also possible to use SMC sampler output to construct an alternative path sampling estimate of the normalising constant (Zhou *et al.*, 2015).
- It is not usually necessary to normalise the weights at each step; here we have included this to simplify the presentation of the algorithm.

### 2.1.2 Increasing dimension

We now describe a case where the parameter  $\theta$  increases in dimension with the number of SMC iterations. Our approach is to set up an SMC sampler on an extended space that has the same dimension of the maximum dimension of  $\theta$  that we will consider. To describe the algorithm accurately we require some additional notation. At SMC iteration  $t$ , we use:  $\theta_t$  to denote the random vector of interest;  $u_t$  to denote a random vector that contains the additional dimensions added to the parameter space at iteration  $t + 1$ , and  $v_t$  to denote the remainder of the dimensions that will be required at future iterations. Our SMC sampler is constructed on a sequence of distributions  $\varphi_t$  of the random vector  $\vartheta_t = (\theta_t, u_t, v_t)$  in space  $E$ . The distribution  $\varphi_t$  is defined to be

$$\varphi_t(\vartheta_t) = \pi_t(\theta_t) \psi_t(u_t | \theta_t) \phi_t(v_t | \theta_t, u_t), \quad (5)$$

where  $\pi_t$  is the distribution of interest at iteration  $t$ , and  $\psi_t$  and  $\phi_t$  are (normalised) distributions on the additional variables. Note that  $\pi_t$  is a marginal distribution of  $\varphi_t$  and since  $\psi_t$  and  $\phi_t$  are normalised,  $\pi_t$  and  $\varphi_t$  have the same normalising constant. With this construction, the weight update the SMC sampler is

$$\begin{aligned} \tilde{w}_{t+1}^{(p)} &= w_t^{(p)} \frac{\varphi_{t+1}(\vartheta_t^{(p)})}{\varphi_t(\vartheta_t^{(p)})} \\ &= w_t^{(p)} \frac{\pi_{t+1}(\theta_t^{(p)}, u_t^{(p)}) \phi_t(v_t^{(p)} | \theta_t^{(p)}, u_t^{(p)})}{\pi_t(\theta_t^{(p)}) \psi_t(u_t^{(p)} | \theta_t^{(p)}) \phi_t(v_t^{(p)} | \theta_t^{(p)}, u_t^{(p)})} \\ &= w_t^{(p)} \frac{\pi_{t+1}(\theta_t^{(p)}, u_t^{(p)})}{\pi_t(\theta_t^{(p)}) \psi_t(u_t^{(p)} | \theta_t^{(p)})}. \end{aligned} \quad (6)$$

We note that this weight update involves none of the dimensions above  $t + 1$ . Our MCMC move must have target  $\varphi_{t+1}$ , starting from  $\vartheta_t^{(p)}$  and storing the result in  $\vartheta_{t+1}^{(p)}$ . We note that if  $\phi_t$  are chosen such that we may simulate from them directly, then our move on  $v_t$  may simply be  $v_t \sim \phi_t$ . However, we note that at the next iteration, only a subvector of  $v_t$  will be involved in the weight update (the variables that correspond to  $u_{t+1}$ ), thus it is only this subvector that we need to simulate in practice.

We note that the auxiliary variables  $v_t$  additional to the target  $\pi_{t+1}$  end up having no impact on the algorithm up until the point when they are required in order to “fill in” additional dimensions. This has the consequence that the choice of a specific maximum size of the space of  $\vartheta_t$  is a technical construct that is not required in practice. This algorithm yields an empirical approximation to  $\pi_t$ , this being a marginal of the empirical approximation to  $\varphi_t$ , and an estimate of  $Z_t$ . These estimates do not suffer additional Monte Carlo variance due to the additional auxiliary variables in  $\varphi_t$ .

## 2.2 Motivating example: Gaussian mixture models

### 2.2.1 Gaussian mixture models

This following sections introduce the novel contributions in the paper, in which we make use of transformations and other ideas in order to improve the efficiency of the sampler. To motivate this, we consider the case of Bayesian model comparison, in which the  $\pi_t$  are different models ordered by their complexity. In section 4 we present an application to Gaussian mixture models, and we use this as our motivating example here. We consider mixture models with  $k$  components, to be estimated from data  $y$ , consisting of  $N$  observed data points. For simplicity we describe a “without completion” model, where we do not introduce a label  $z$  that assigns data points to components (see Jasra *et al.* (2005b) for a comprehensive description of Bayesian inference of Gaussian mixtures). Let the  $s$ th component have a mean  $\mu_s$ , precision  $\tau_s$  and weight  $\nu_s$ , with the weights summing to one over the components, let

$p_\mu$  and  $p_\tau$  be the respective priors on these parameters, which are the same for every component, and let  $p_\nu$  be the joint prior over all of the weights. The likelihood under  $k$  components is

$$f_t(y | \theta_k = (\mu_s, \tau_s, \nu_s)_{s=1}^k) = \prod_{i=1}^N \sum_{s=1}^k \nu_s \mathcal{N}(y_i | \mu_s, \tau_s^{-1}), \quad (7)$$

where  $\mathcal{N}$  is the normal density with the specified mean and variance.

### 2.2.2 RJMCMC for Gaussian mixture models

An established approach for estimating mixture models is that of RJMCMC. Here,  $k$  is chosen to be a random variable and assigned a prior  $p_k$ , which here we choose to be uniform over the values 1 to  $T$ . Let

$$\pi_k(\theta_k) = \pi(\theta_k | t, y) \propto p_\nu(\nu_{1:k}) \left( \prod_{s=1}^t p_\mu(\mu_s) p_\tau(\tau_s) \right) f_k(y | \theta_k = (\mu_s, \tau_s, \nu_s)_{s=1}^k) \quad (8)$$

be the joint posterior distribution over the parameters  $\theta_k$  conditional on  $k$ . RJMCMC is an approach to simulating from the joint space of  $(k, \theta_k)$  in which a mixture of moves is used, some fixed-dimensional (to explore within a model without changing  $k$ ) and some trans-dimensional (to explore different values of  $k$ ). The simplest type of trans-dimensional move in this example is that of a birth move for moving from  $k$  to  $k+1$  components, or a death move for moving from  $k+1$  to  $k$  (Richardson and Green, 1997). Here we consider a birth move, and for the purposes of exposition we assume that the weights of the components are chosen to be fixed in each model (this assumption will be relaxed later).

Let  $u_k = (\mu_{k+1}, \tau_{k+1})$ , be the mean and precision of the new component and let  $\psi_k(u_k | \theta_t) = p_\mu(\mu_{k+1}) p_\tau(\tau_{k+1})$ . A birth move simulates  $u_k \sim \psi_k$  and has acceptance probability

$$\alpha = \min \left\{ 1, \frac{\pi_{k+1}(\theta_{k+1})}{\pi_k(\theta_k)} \frac{p_k(k+1) q(k | k+1)}{p_k(k) q(k+1 | k)} \right\}, \quad (9)$$

where the ratio of the  $q$ s accounts for the relative probability of using death and birth moves.

### 2.2.3 Linking RJMCMC and the SMC sampler with increasing dimension

Now consider the use of an SMC sampler for inference where the sequence of target distributions is  $(\pi_t)_{t=1}^T$ , i.e. the  $t$ th distribution is the mixture of Gaussians with  $t$  components. By choosing  $u$  and  $\psi$  as above, the notation here makes clear that the distributions used in this example have exactly same character as those in section 2.1.2, and thus that we may use the SMC sampler described in that section. To complete the specification required for this SMC sampler, we also choose  $v_t = (\mu_{(t+2):T}, \tau_{(t+2):T})$  and  $\psi_t(v_t | \theta_t, u_t) = \prod_{s=t+2}^T p_\mu(\mu_s) p_\tau(\tau_s)$ . The output of this algorithm is an approximation  $\hat{\pi}_t$  of each distribution, and an estimate  $\hat{Z}_t$  of its marginal likelihood (which can be multiplied by  $p_t$  to obtain the posterior model probability).

We note that this output is very similar to the output of RJMCMC, which yields an approximation  $\hat{\pi}_t$  of each distribution, and estimates of the posterior model probability, and note the strong similarity of acceptance probability in equation 9 to the SMC weight in equation 6. In fact, taking the view of RJMCMC in Karagiannis and Andrieu (2013) we observe that where  $\theta_t \sim \pi_t$ ,  $u_t \sim \psi_t$  and  $\theta_{t+1} = (\theta_t, u_t)$ , an IS estimator of the ratio  $Z_{t+1}/Z_t$  is given by

$$\frac{\widehat{Z}_{t+1}}{Z_t} = \frac{\pi_{t+1}(\theta_{t+1})}{\pi_t(\theta_t) \psi_t(u_t | \theta_t)}. \quad (10)$$

Thus we may see RJMCMC as using an IS estimator of the ratio of the posterior model probabilities within its acceptance ratio; this view on RJMCMC links it to pseudo-marginal approaches (Andrieu and Roberts, 2009) in which IS estimators of target distributions are employed. As in pseudo-marginal MCMC, the efficiency of the chain depends on the variance of the estimator that is used. We observe that the IS estimator in equation 10 is likely to have high variance: this is one way of explaining the poor acceptance rate of dimension changing moves in RJMCMC.

This view suggests a number of potential improvements to RJMCMC with a birth move, each of which has been previously investigated.

- IS performs better if the proposal distribution (the denominator) is close to the target distribution, whilst ensuring that the proposal has heavier tails than the target. The birth move yields an IS estimator in which the proposal is very likely to have heavy tails, since the prior is often used for  $\psi_t$ . However, in many cases it is far from the target. The original RJMCMC algorithm addresses this by allowing for the use of transformations to move from the parameters of one model to the parameters of another. Richardson and Green (1997) provide a famous example of this in the Gaussian mixture case in the form of split-merge moves. Focussing on the split move, the idea is to propose splitting an existing component, using a moment matching technique to ensure that the new components have appropriate means, variances and weights.
- Annealed importance sampling (AIS) (Neal, 2001) provides estimates that are a low variance alternative to IS estimates. The idea is to use intermediate distributions to form a path between the IS proposal and target, using MCMC moves to move points along this path. This approach was shown to be beneficial in some cases by Karagiannis and Andrieu (2013).
- The estimator in equation 10 uses only a single importance point. It would be improved by using multiple points. However, using such an estimator directly within RJMCMC leads to a “noisy” algorithm that does not have the correct target distribution for the same reasons as those given for the noisy exchange algorithm in Alquier *et al.* (2016).

The approach we take in this paper is to investigate variations on these ideas within the SMC sampler context, rather than RJMCMC. We begin by examining the use of transformations in section 2.3, then describe the use of intermediate distributions and other refinements in section 2.4. The final idea is automatically used in the SMC context, due to the use of  $P$  particles.

## 2.3 Using transformations in SMC samplers

In this section we outline how to use transformations within SMC; an approach we will refer to as *transformation SMC* (TSMC). TSMC is a generalisation of the approach described in section 2.1.2. Here we again use the approach of performing SMC on a sequence of targets  $\varphi_t$ , with each of these targets being on a space of fixed dimension, constructed such that they have the desired target  $\pi_t$  as a marginal. In this section the dimension of the space on which  $\pi_t$  is defined again varies with  $t$ , but is not necessarily increasing with  $t$ . We use a SMC sampler in the case where the space  $E_t$  may change with  $t$ , as described in Del Moral *et al.* (2007).

Following notation similar to section 2.1.2, we let  $\theta_t$  be the random vector of interest at SMC iteration  $t$ . As before, we are interested in approximating the distributions  $\pi_t$  of  $\theta_t$  in the space  $\Theta_t$ . To set up the SMC, we use a sequence of unnormalised targets  $\varphi_t$  (whose normalised versions are  $\bar{\varphi}_t$ ), being the distribution of the random vector  $\vartheta_t = (\theta_t, u_t)$  in the space  $E_t = (\Theta_t, U_t)$  and use

$$\varphi_t(\theta_t, u_t) = \pi_t(\theta_t) \psi_t(u_t | \theta_t).$$

The dimension of  $\Theta_t$  can change with  $t$ , but the dimension of  $E_t$  must be constant in  $t$ . Using notation similar to that of Karagiannis and Andrieu (2013), we introduce a transformation  $G_{t \rightarrow t+1} : \Theta_t \times U_t \rightarrow \Theta_{t+1} \times U_{t+1}$  and define

$$\begin{aligned} \vartheta_{t \rightarrow t+1} &:= G_{t \rightarrow t+1}(\vartheta_t), \\ (\theta_{t \rightarrow t+1}(\vartheta_t), u_{t \rightarrow t+1}(\vartheta_t)) &:= G_{t \rightarrow t+1}(\vartheta_t), \end{aligned}$$

In many cases we will choose  $G_{t \rightarrow t+1}$  to be bijective. In this case we denote its inverse by  $G_{t+1 \rightarrow t} = G_{t \rightarrow t+1}^{-1}$ , with

$$\begin{aligned} \vartheta_{t+1 \rightarrow t} &:= G_{t+1 \rightarrow t}(\vartheta_{t+1}), \\ (\theta_{t+1 \rightarrow t}(\vartheta_{t+1}), u_{t+1 \rightarrow t}(\vartheta_{t+1})) &:= G_{t+1 \rightarrow t}(\vartheta_{t+1}). \end{aligned}$$

Let the distribution of the transformed random variable  $\vartheta_{t \rightarrow t+1}$  be  $\varphi_{t \rightarrow t+1}$  (i.e.  $\bar{\varphi}_{t \rightarrow t+1} = \mathcal{L}(\vartheta_{t \rightarrow t+1}) = \mathcal{L}(G_{t \rightarrow t+1}(\vartheta_t))$  where  $\mathcal{L}(X)$  denotes the law of a random variable  $X$ ) and let the distribution of  $\vartheta_{t+1 \rightarrow t}$  be  $\varphi_{t+1 \rightarrow t}$ . These distributions may be derived using standard results about the distributions of transforms of random variables. For example, consider the case where the  $E_t$  are continuous spaces and where  $G_{t \rightarrow t+1}$  is a diffeomorphism, having Jacobian determinant  $J_{t \rightarrow t+1}$ , and where the inverse  $G_{t+1 \rightarrow t}$  has Jacobian determinant  $J_{t+1 \rightarrow t}$ . In this case we have that

$$\begin{aligned} \varphi_{t \rightarrow t+1}(\vartheta_{t \rightarrow t+1}) &= \varphi_t(G_{t+1 \rightarrow t}(\vartheta_{t \rightarrow t+1})) |J_{t+1 \rightarrow t}| \\ &= \pi_t(\theta_{t+1 \rightarrow t}(\vartheta_{t \rightarrow t+1})) \psi_t(u_{t+1 \rightarrow t}(\vartheta_{t \rightarrow t+1}) | \theta_{t+1 \rightarrow t}(\vartheta_{t \rightarrow t+1})) |J_{t+1 \rightarrow t}|. \end{aligned}$$

$$\begin{aligned}
\varphi_{t+1 \rightarrow t}(\vartheta_{t+1 \rightarrow t}) &= \varphi_{t+1}(G_{t \rightarrow t+1}(\vartheta_{t+1 \rightarrow t})) |J_{t \rightarrow t+1}| \\
&= \pi_{t+1}(\theta_{t \rightarrow t+1}(\vartheta_{t+1 \rightarrow t})) \psi_{t+1}(u_{t \rightarrow t+1}(\vartheta_{t+1 \rightarrow t}) | \theta_{t \rightarrow t+1}(\vartheta_{t+1 \rightarrow t})) |J_{t \rightarrow t+1}|.
\end{aligned}$$

We may then use an SMC sampler on the sequence of targets  $\varphi_t$ , with the following steps at its  $(t+1)$ th iteration.

1. **Transform:** For the  $p$ th particle,, apply  $\vartheta_{t \rightarrow t+1}^{(p)} = G_{t \rightarrow t+1}(\vartheta_t^{(p)})$ .
2. **Reweight:** Calculate the updated (unnormalised) weight  $\tilde{w}_{t+1}^{(p)}$  of the  $p$ th particle  $\vartheta_t^{(p)}$

$$\tilde{w}_{t+1}^{(p)} = w_t^{(p)} \frac{\varphi_{t+1}(\vartheta_{t \rightarrow t+1}^{(p)})}{\varphi_{t \rightarrow t+1}(\vartheta_{t \rightarrow t+1}^{(p)})}. \quad (11)$$

Where  $G_{t \rightarrow t+1}$  is a diffeomorphism we may express the weight update as either

$$\tilde{w}_{t+1}^{(p)} = w_t^{(p)} \frac{\pi_{t+1}(\theta_{t \rightarrow t+1}(\vartheta_t^{(p)})) \psi_{t+1}(u_{t \rightarrow t+1}(\vartheta_t^{(p)}) | \theta_{t \rightarrow t+1}(\vartheta_t^{(p)})) |J_{t \rightarrow t+1}|}{\pi_t(\theta_t^{(p)}) \psi_t(u_t^{(p)} | \theta_t^{(p)})},$$

when working in the space  $E_t$ , or equivalently

$$\tilde{w}_{t+1}^{(p)} = w_t^{(p)} \frac{\pi_{t+1}(\theta_{t \rightarrow t+1}^{(p)}) \psi_{t+1}(u_{t \rightarrow t+1}^{(p)} | \theta_{t \rightarrow t+1}^{(p)})}{\pi_t(\theta_{t+1 \rightarrow t}(\vartheta_{t \rightarrow t+1}^{(p)})) \psi_t(u_{t+1 \rightarrow t}(\vartheta_{t \rightarrow t+1}^{(p)}) | \theta_{t+1 \rightarrow t}(\vartheta_{t \rightarrow t+1}^{(p)})) |J_{t+1 \rightarrow t}|}$$

when working in the space  $E_{t+1}$ . These two weight updates give identical values: it is straightforward to see that both are equal to

$$\tilde{w}_{t+1}^{(p)} = w_t^{(p)} \frac{\pi_{t+1}(\theta_{t \rightarrow t+1}^{(p)}) \psi_{t+1}(u_{t \rightarrow t+1}^{(p)} | \theta_{t \rightarrow t+1}^{(p)})}{\pi_t(\theta_t^{(p)}) \psi_t(u_t^{(p)} | \theta_t^{(p)}) |J_{t+1 \rightarrow t}|}. \quad (12)$$

The weight update in equation 6 is a special case of this in which the identity transformation is used. We note that again it is possible, depending on the transformation used, that this weight update involves none of the dimensions above  $\max\{\dim(\theta_t), \dim(\theta_{t+1})\}$ .

3. **Resample.** As previously.

4. **Move.** For each particle perform an MCMC move with target  $\varphi_{t+1}$ , starting from  $\vartheta_{t \rightarrow t+1}^{(p)}$  and storing the result in  $\vartheta_{t+1}^{(p)}$ . As previously, it is possible that we may avoid the simulation of  $u$  variables that are not used at the next iteration.

To illustrate the additional flexibility this framework allows, over and above the sampler described in section 2.1.2, we return to considering the Gaussian mixture example in section 2.2. The sampler from 2.1.2 provides an alternative to RJMCMC in which a set of particles is used to sampler from each model in turn, using the particles from model  $t$ , together with new dimensions simulated using a birth move, to explore model  $t+1$ . The sampler in this section allows us to use a similar idea using more sophisticated proposals, in particular we may use split moves (and also tackle the case of birth moves where the component weights are not fixed).

In general, this sampler provides a very flexible way to move points between models. The efficiency of the sampler depends on the choice of  $\psi_t$ , which may be any distribution which we may simulate and evaluate pointwise and  $G_{t \rightarrow t+1}$ . As previously, a good choice for these quantities should result in a small distance between  $\varphi_{t \rightarrow t+1}$  and  $\varphi_{t+1}$ : the effect of this on the theoretical properties of the sampler is discussed in section 2.6. However, we also desire that  $\varphi_{t \rightarrow t+1}$  has heavier tails than  $\varphi_{t+1}$ . As in the design of RJMCMC algorithms, usually these choices will usually be designed using insight into the particular model under consideration.

We note that an equation similar to equation 11 is used in Dinh *et al.* (2016) for the special case of discrete distributions on trees. However, this weight is used in the context of constructing an SMC sampler that uses birth moves for additional dimensions, as is presented in section 2.1.2.

## 2.4 Design of SMC samplers

### 2.4.1 Using intermediate distributions

The Monte Carlo variance of an SMC sampler depends on the distance between successive target distributions (section 2.6), thus a well designed sampler will use a sequence of distributions in which the distance between successive distributions is not large. This is particularly clear in the case of using an MCMC move as the kernel, where the weight update in equation 1 results from choosing the SMC backwards kernel to be the reverse MCMC move, which is only close to optimal when the distance between successive distributions is small. We see that from the weight update that even if MCMC moves that mix perfectly (i.e. they simulate exactly from  $\pi_{t+1}$ ), a large distance between  $\pi_t$  and  $\pi_{t+1}$  yields a degenerate sampler. In the applications described in section 1.2 it is possible that the distance between successive targets is large, therefore we propose to modify the algorithms described above by introducing intermediate distributions in between successive targets: in between targets  $\varphi_t$  and  $\varphi_{t+1}$  we use  $K-1$  intermediate distributions, the  $k$ th being  $\varphi_{t,k}$ , so that  $\varphi_{t,0} = \varphi_t$  and  $\varphi_{t,K} = \varphi_{t+1}$  and therefore  $\varphi_{t,K} = \varphi_{t+1,0}$ . This idea is well known in the SMC context (e.g. Beskos *et al.* (2014)) and is also used in the RJMCMC context by Karagiannis and Andrieu (2013), in which the motivation is improving the IS estimate used in RJMCMC (equation 10) using AIS.

Our notation is closely related to this latter paper. Karagiannis and Andrieu (2013) describes two alternative forms for the intermediate distributions.

1. *Geometric annealing* uses the sequence

$$\varphi_{t,k}(\vartheta_{t,k}) = [\varphi_{t+1 \rightarrow t}(\vartheta_{t,k})]^{\gamma_k} [\varphi_t(\vartheta_{t,k})]^{1-\gamma_k}, \quad (13)$$

where  $0 = \gamma_0 < \gamma_1 < \dots < \gamma_K = 1$ . We may also write this as a distribution  $\varphi_{t \rightarrow t+1,k}(\vartheta_{t \rightarrow t+1,k})$  on  $E_{t+1}$

$$\varphi_{t \rightarrow t+1,k}(\vartheta_{t \rightarrow t+1,k}) = [\varphi_{t+1}(\vartheta_{t \rightarrow t+1,k})]^{\gamma_k} [\varphi_{t \rightarrow t+1}(\vartheta_{t \rightarrow t+1,k})]^{1-\gamma_k}, \quad (14)$$

2. *Arithmetic annealing* uses

$$\varphi_{t,k}(\vartheta_{t,k}) \propto \gamma_k [\varphi_{t+1 \rightarrow t}(\vartheta_{t,k})] + (1 - \gamma_k) [\varphi_t(\vartheta_{t,k})], \quad (15)$$

where  $0 = \gamma_0 < \gamma_1 < \dots < \gamma_K = 1$ , which is equivalent to

$$\varphi_{t \rightarrow t+1,k}(\vartheta_{t \rightarrow t+1,k}) \propto \gamma_k [\varphi_{t+1}(\vartheta_{t \rightarrow t+1,k})] + (1 - \gamma_k) [\varphi_{t \rightarrow t+1}(\vartheta_{t \rightarrow t+1,k})]. \quad (16)$$

This idea results in only small alterations to the TSMC presented above. We now use a sequence of targets  $\varphi_{t,k}$ , incrementing the  $t$  index then using a transform move  $\vartheta_{t \rightarrow t+1,0}^{(p)} = G_{t \rightarrow t+1}(\vartheta_{t,K}^{(p)})$  each time  $k = 0$ . The weight update becomes

$$\tilde{w}_{t+1,k+1}^{(p)} = w_{t+1,k}^{(p)} \frac{\varphi_{t \rightarrow t+1,k+1}(\vartheta_{t \rightarrow t+1,k}^{(p)})}{\varphi_{t \rightarrow t+1,k}(\vartheta_{t \rightarrow t+1,k}^{(p)})}, \quad (17)$$

and the MCMC moves now have target  $\varphi_{t \rightarrow t+1,k+1}$ , starting from  $\vartheta_{t \rightarrow t+1,k}^{(p)}$  and storing the result in  $\vartheta_{t \rightarrow t+1,k+1}^{(p)}$ . The use of intermediate distributions makes this version of TSMC more robust than the previous one; the MCMC moves used at the intermediate distributions provide a means for the the algorithm to recover if the initial transformation is not enough to ensure that  $\varphi_{t \rightarrow t+1}$  is similar to  $\varphi_{t+1}$ .

Throughout the remainder of the paper we use this form of the algorithm, in which our particles are transformed to the space  $E_{t+1}$  at the beginning of the transition through the intermediate distributions. However we note that in some circumstances it may be more efficient to design MCMC moves to explore the space  $E_t$ : in this case the reweighting equation would be given in terms of  $\varphi_{t,k}(\vartheta_{t,k})$ ; the move step would use MCMC moves with target  $\varphi_{t,k+1}$ , starting from  $\vartheta_{t,k}^{(p)}$  and storing the result in  $\vartheta_{t,k+1}^{(p)}$ ; and the transform would be used after the move step at  $k = K$  rather than at the first step.

### 2.4.2 Multiple routes between targets

Returning to the mixture of Gaussians example, we now note a limitation in the approach we have described thus far. In the SMC sampler we use, we make use of a pre-specified route through a number of models, using either



birth or split moves to move to a model with more components. However, in the case of mixtures, for two or more components, when using a split move we require to choose the component that is to be split. Our current algorithm only allows for one component to be split, which may lead to suboptimal performance if this choice is inappropriate. Note that this problem is not encountered in RJMCMC, where the choice of which component to split is made many times.

We may think of the choice of splitting different components as offering multiple “routes” through a space of distributions, with the same start and end points. Another alternative route would be given by using a birth move rather than a split move. In this section we generalise TSMC to allow multiple routes. We restrict our attention to the case where the choice of multiple routes is possible at the beginning of a transition from  $\varphi_t$  to  $\varphi_{t+1}$ , when  $k = 0$  (more general schemes are possible). Let  $r_{t+1}$  be a discrete random variable, which has  $R$  states, one for each possible route. A route corresponds to a particular choice for the transformation  $G_{t \rightarrow t+1}$ , thus we consider a set of  $R$  possible transformations indexed by  $r_t$ , using the notation  $G_{t \rightarrow t+1}^{(r_t)}$  (also using this superscript on distributions that depend on this choice of  $G$ ). Similar to a related approach in Del Moral *et al.* (2006) we now augment the target distribution with variables  $r_0, \dots, r_{T-1}$  with distributions  $\rho_0, \dots, \rho_{T-1}$ , with these variables being independent of each other and of the other variables in the target. Our sampler will use proposals  $\rho_0, \dots, \rho_{T-1}$  for these variables at the point at which they are introduced, so that different particles use different routes, but will not perform any MCMC moves on the variable after it is introduced. This leads to the sampler being degenerate in most of the  $r$  variables, but this doesn’t affect the desired target distribution.

The revised form of TSMC is then, when  $k = 0$ , to first simulate routes  $r_t^{(p)} \sim \rho_t$  for each particle, then to use a different transform  $\vartheta_{t \rightarrow t+1,0}^{(p)} = G_{t \rightarrow t+1}^{(r_t^{(p)})}(\vartheta_{t,K}^{(p)})$  dependent on the route variable. The weight update is then given by

$$\tilde{w}_{t+1,k+1}^{(p)} = w_{t+1,k}^{(p)} \frac{\varphi_{t \rightarrow t+1,k+1}^{(r_t^{(p)})}(\vartheta_{t \rightarrow t+1,k}^{(p)})}{\varphi_{t \rightarrow t+1,k}^{(r_t^{(p)})}(\vartheta_{t \rightarrow t+1,k}^{(p)})}, \quad (18)$$

with the weight update on space  $E_{t+1}$ , and the MCMC move has target  $\varphi_{t \rightarrow t+1,k+1}^{(r_t^{(p)})}$ .

### 2.4.3 Adapting the sequence of intermediate distributions

Section 2.4.1 describes the use of intermediate distributions with the aim of ensuring that the distance between neighbouring targets is not too great, but this aim cannot be achieved without also considering where to place these intermediate distributions. There is no general answer to this question, thus in this paper we follow the adaptive strategy first used in Del Moral *et al.* (2012) and refined in Zhou *et al.* (2015) in the case where resampling is not performed at every iteration. At iteration  $(t + 1)$ ,  $(k + 1)$  this approach uses the conditional ESS (CESS)

$$\text{CESS} = \frac{P \left( \sum_{p=1}^P w_{t+1,k}^{(p)} \omega^{(p)} \right)^2}{\sum_{p=1}^P w_{t+1,k}^{(p)} (\omega^{(p)})^2}, \quad (19)$$

to monitor the discrepancy between neighbouring distributions, where  $\omega^{(p)}$  is the incremental weight (i.e. the term multiplied by  $w_{t+1,k}^{(p)}$  in equation 17). Before the reweighting step is performed, the next intermediate distribution is chosen to be the distribution under which the CESS is found to be  $\beta P$ , for some  $0 < \beta < 1$ . In the case of the geometric or arithmetic annealing schemes described above, this corresponds to a particular choice for  $\gamma_k$ .

### 2.4.4 Adapting MCMC proposal distributions

The use of an SMC sampler allows us much freedom in the adaptation of the MCMC kernels used to explore the space based on the past history of the sampler; this is a contrast to MCMC, in which adaptation must be performed with care (Andrieu and Thoms, 2008). Several strategies have been developed (e.g. Fearnhead and Taylor (2013)), but a straightforward scheme for adaptation that we adopt here is to use the sample variance of the current particle set in order to set the variance of the proposals used in Metropolis-Hastings moves (Beaumont *et al.* (2009) uses twice the sample variance). We note that since we often use single component (Gibbs-type) moves in our MCMC kernels this choice may not always be appropriate; the sample variance yielding an estimate of the marginal posterior variance rather than the desired (smaller) conditional variance. However, in practice we often find this strategy to work well, and in this paper we directly use the marginal sample variance as the proposal variance.

## 2.5 Comparison to other model comparison techniques

One of the most obvious applications of TSMC is Bayesian model comparison. In this section we provide a comparison to existing techniques. Zhou *et al.* (2015) provides an excellent overview of current approaches, with a particular focus on SMC. Broadly speaking there are three classes of methods: those that estimate the marginal likelihood separately (e.g. the harmonic mean estimator (Raftery *et al.*, 2006)); others that estimate Bayes' factors between pairs of models (e.g. bridge/path sampling (Gelman, 1998)); and those that explore the joint space of models and parameters (e.g. RJMCMC). Methods in the first two classes are well suited to cases in which there are relatively few models, whereas methods in the third class can (in theory) explore a countable set of models, although are likely not to estimate the marginal likelihood of lower probability models accurately. Methods in the latter two classes usually exploit nearby models being close to each other; using points sampled from one model to explore regions of high posterior mass in a neighbouring model.

This paper describes a method in the second class. An SMC algorithm moving from  $\varphi_t$  to  $\varphi_{t+1}$  estimates the ratio of the normalising constants of these distributions. We describe a sequential use of this approach that is particularly suited where there exists an ordering of the models such that neighbouring models are similar to each other. In this section we provide a discussion of how our approach compares to two approaches in particular: the first being RJMCMC, and in particular the AIS RJMCMC variant; the second being an SMC sampler approach in Zhou *et al.* that has some similarities with our proposed method.

### 2.5.1 AIS-RJMCMC

As previously described, RJMCMC is an MCMC algorithm that explores the joint space of models and parameters. At some iterations, a model-changing move is made; equation 10 shows how we may think of this move as using a single point IS estimator of the marginal likelihood. The high variance of this estimator can lead to the efficiency of RJMCMC being poor. Karagiannis and Andrieu (2013) suggests the use of a (more accurate) AIS estimator instead, using intermediate distributions of the style we describe in section 2.4.1, and this is shown to improve efficiency.

TSMC uses the alternative approach of an SMC sampler moving from the first model in a sequence to the last, estimating the posterior distribution of each model and its marginal likelihood along the way. In comparing TSMC and AIS-RJMCMC, the usual comparisons between SMC and MCMC apply: i.e. it is easier to use adaptive methods in SMC and SMC has no burn in; however SMC uses a fixed population of particles and cannot be run for an arbitrarily long time to achieve accurate results as MCMC can. However, there also are two points specific to this situation.

Firstly, to move between models, a population of particles is used in the SMC sampler, compared to a single point when using RJMCMC. This means that each individual model-changing move in the RJMCMC will estimate the ratio of normalising constants between the models less accurately than the SMC. There is an additional implication in that in the SMC the use of a population enables intermediate distributions between the two models to be chosen adaptively, as in section 2.4.3, which is not possible when using a single point. This is potentially important: without adaptation it can be very difficult to choose a suitable sequence of intermediate distributions. Karagiannis and Andrieu (2013) outlines problems with non-adaptive schemes for both the geometric and arithmetic annealing cases, pointing out that very many intermediate distributions may be required to create a smooth path between the end points. When using geometric annealing, when a proposed  $\vartheta_{t \rightarrow t+1,k}$  has very low (or zero) probability under the model  $\varphi_{t+1}$ , it will also have very low (or zero) probability in all intermediate distributions except when  $\gamma_k$  is very small. When used in TSMC, if this is the case for all particles, the algorithm may suffer numerical problems due to all particles being assigned a weight close to (or exactly) zero. Arithmetic annealing would appear to reduce this possibility, but suffers the drawback that the two models are not simply weighted by  $\gamma_k$  and  $(1 - \gamma_k)$  as they appear to be in equations 15 and 16. Rather, the normalising constants (i.e. the unknown marginal likelihoods) of the two end point models result in a different scaling for the two models, making it unclear as to how to optimally space the  $\gamma_k$ . When using TSMC where  $\gamma_k$  is chosen adaptively, we avoid this issue.

Secondly, TSMC only ever makes single transition from model  $\varphi_t$  to  $\varphi_{t+1}$ , whereas RJMCMC makes many moves, including moving back from  $\varphi_{t+1}$  to  $\varphi_t$ . This puts a requirement on TSMC that the single transition it makes must be successful. However, it creates an issue for RJMCMC, which we may see through again using the IS interpretation in equation 10. Recall that we desire the denominator in this expression to have heavier tails than the numerator in order to ensure that it is an accurate IS estimator. However, this requirement means that when making a move from  $\varphi_{t+1}$  to  $\varphi_t$ , we result in the denominator in the corresponding IS estimator having lighter tails than the numerator. Thus we see that unless the neighbouring distributions are very similar, we make use of a low quality IS estimator when moving either up or down in dimension. The use of AIS instead of IS helps to minimise

this problem, but it does not occur at all in TSMC, where the move is only made in one direction.

### 2.5.2 SMC3

Zhou *et al.* (2015) introduce several alternative SMC approaches to model selection; one in each of the classes of algorithms mentioned previously. Jasra *et al.* (2008) also introduces an alternative approach to efficiently exploring the joint model-parameter posterior. The characteristics of these approaches is described in Zhou *et al.* (2015). Here we focus on the algorithm they call SMC3, which is closest to TSMC in that it is also an SMC sampler that transitions from a model  $\pi_t$  to another  $\pi_{t+1}$ . However, the sampler is set up in a different way; each particle lies in the joint space of the two models and their parameters (the same space as RJMCMC would use when there are two models), rather than in the parameter space of the higher dimensional model used in TSMC. Every member of the sequence of distributions is on the joint model-parameter space, and the sequence uses an artificial prior distribution in which the initial prior is 1 for model  $\pi_t$  and 0 for model  $\pi_{t+1}$ , which changes gradually through the sequence such that all of the prior mass is on model  $\pi_{t+1}$ . The SMC sampler then, at iteration  $(k + 1)$ , has a weight update of

$$\tilde{w}_{t+1,k+1}^{(p)} = w_{t+1,k}^{(p)} \frac{p_{k+1} \left( m_k^{(p)} \right)}{p_k \left( m_k^{(p)} \right)},$$

where  $p_k$  is the artificial model prior at distribution  $k$ , and uses an RJMCMC move on each particle where the model prior is  $p_{k+1}$ .

This approach has some advantages over TSMC, but also some disadvantages. Firstly, it is straightforward to code up, and it has the advantage that the weights will not be very variable since  $p_k$  is likely to be very close to  $p_{k+1}$ . However, it inherits the disadvantages of RJMCMC; mainly that it may be difficult to design these moves so that that are efficient. It would be possible to use AIS-RJMCMC moves as an alternative, with the same advantages and disadvantages holding in this context as in the previous section. Ultimately, we expect TSMC to be well suited to some problems, and SMC3 to others.

One final remark about SMC3 is that it is not completely straightforward to use an adaptive method to choose the sequence of distributions, since all of the particles begin in a single model, thus at the first weight update all of the incremental weights are the same (hence the CESS is the same) regardless of the choice of  $p_{k+1}$ . One can envisage pragmatic schemes that avoid this issue.

## 2.6 Theoretical aspects

Using the notation introduced in section 2.3, we define a sequence of (unnormalised) targets  $\{\varphi_{t \rightarrow T}\}_{t=0}^T$  on the same measurable space  $(E_T, \mathcal{E}_T)$  as follows. Consider  $\vartheta_t \sim \varphi_t(\cdot)$  and define

$$\bar{\varphi}_{t \rightarrow T}(\cdot) := \mathcal{L}(G_{t \rightarrow T}(\vartheta_t)),$$

where  $\mathcal{L}(X)$  denotes the law of a random variable  $X$ , and where  $G_{t \rightarrow T}$  is defined recursively for  $0 \leq t < T$  as follows

$$G_{t \rightarrow T} := G_{t+1 \rightarrow T} \circ G_{t \rightarrow t+1},$$

with  $G_{T \rightarrow T}$  as the identity function. Notice that the normalising constant associated to  $\varphi_{t \rightarrow T}$  is equal to  $Z_t$ . Hence, the TSMC algorithm described in section 2.3 can be seen as an SMC sampler for the targets  $\{\varphi_{t \rightarrow T}\}_t$ , propagating particles  $\left\{ \vartheta_{t \rightarrow T}^{(p)} \right\}_{t,p}$  using a sequence of MCMC kernels  $\{K_{t \rightarrow T} : E_T \times \mathcal{E}_T \rightarrow [0, 1]\}_t$ , where  $K_{t \rightarrow T}$  admits  $\varphi_{t \rightarrow T}$  as invariant. This will also be the case for the modified TSMC algorithm discussed in section 2.4.1, however the details are omitted for simplicity.

Therefore, after the  $(t + 1)$ th iteration the target  $\varphi_{t+1 \rightarrow T}$  can be approximated using

$$\hat{\varphi}_{t+1 \rightarrow T}^P = \sum_{p=1}^P w_{t+1 \rightarrow T}^{(p)} \delta_{\vartheta_{t+1 \rightarrow T}^{(p)}},$$

where, for every  $p \in \{1, \dots, P\}$ ,

$$w_{t+1 \rightarrow T}^{(p)} \propto w_{t \rightarrow T}^{(p)} \frac{\varphi_{t+1 \rightarrow T} \left( \vartheta_{t \rightarrow T}^{(p)} \right)}{\varphi_{t \rightarrow T} \left( \vartheta_{t \rightarrow T}^{(p)} \right)}$$

and  $w_{0 \rightarrow T}^{(p)} = 1/P$ . Furthermore, notice that  $w_{t \rightarrow T}^{(p)} = w_t^{(p)}$  for all  $0 \leq t < T$  and any  $p \in \{1, \dots, P\}$ , consequently expectations of the form  $\bar{\varphi}_{t+1}(h)$  (for a function  $h : E_t \rightarrow \mathbb{R}$ ) can be approximated using

$$\begin{aligned}\hat{\varphi}_{t+1 \rightarrow T}^P(h \circ G_{T \rightarrow t+1}) &= \sum_{p=1}^P w_{t+1 \rightarrow T}^{(p)} h \circ G_{T \rightarrow t+1} \left( \vartheta_{t+1 \rightarrow T}^{(p)} \right) \\ &= \sum_{p=1}^P w_{t+1}^{(p)} h \left( \vartheta_{t+1}^{(p)} \right) = \hat{\varphi}_{t+1}^P(h).\end{aligned}$$

The following theorem, found in Del Moral *et al.* (2006, Proposition 2), follows from well-known standard SMC convergence results.

**Theorem 2.1.** *Under weak integrability conditions (see Chopin (2004, Theorem 1) or Del Moral (2004, p300-306)) and for any bounded  $h : E_t \rightarrow \mathbb{R}$ , as  $P \rightarrow \infty$*

1.  $P^{-1/2} \{ \hat{\varphi}_t^P(h) - \bar{\varphi}_t(h) \} \Rightarrow \mathcal{N}(\cdot | 0, \sigma_{IS,t}^2(h))$ , if no resampling is performed;

(a)  $P^{-1/2} \{ \hat{\varphi}_t^P(h) - \bar{\varphi}_t(h) \} \Rightarrow \mathcal{N}(\cdot | 0, \sigma_{SMC,t}^2(h))$ , when multinomial resampling is performed at every iteration;

where  $\sigma_{IS,t}^2(h)$  and  $\sigma_{SMC,t}^2(h)$  follow similar expressions to those in Del Moral *et al.* (2006, Proposition 2).

*Remark 2.1.* As noted also in Del Moral *et al.* (2006), under strong mixing assumptions, the variance  $\sigma_{SMC,t}^2(h)$  can be uniformly bounded in  $t$  whereas  $\sigma_{IS,t}^2(h)$  will typically diverge as  $t$  increases.

Respecting the normalising constants  $\{Z_t\}_{t=1}^T$ , they can be approximated using

$$\hat{Z}_{t+1}^P = \prod_{s=1}^{t+1} \sum_{p=1}^P w_{s \rightarrow T}^{(p)} \frac{\varphi_{s+1 \rightarrow T}(\vartheta_{s \rightarrow T}^{(p)})}{\varphi_{s \rightarrow T}(\vartheta_{s \rightarrow T}^{(p)})} = \prod_{s=1}^{t+1} \sum_{p=1}^P w_s^{(p)} \frac{\varphi_{s+1 \rightarrow s}(\vartheta_s^{(p)})}{\varphi_s(\vartheta_s^{(p)})},$$

and standard results show that these estimates are unbiased (see e.g. Del Moral (2004, proposition 7.4.1)), with relative variance increasing at most linearly in  $t$  Cérou *et al.* (2011, Theorem 5.1). Such results are summarised in the following theorem.

**Theorem 2.2.** *For fixed  $E_T$ , and when resampling is not done adaptively, the estimates  $\{\hat{Z}_t^P\}_t$  satisfy*

$$\mathbb{E} \left[ \hat{Z}_t^P \right] = Z_t.$$

*Furthermore, under strong mixing assumptions there exists a constant  $C_T(t)$ , which is linear in  $t$ , such that*

$$\mathbb{V} \left[ \frac{\hat{Z}_t^P}{Z_t} \right] \leq \frac{C_T(t)}{N}.$$

However, as  $T$  increases the dimension of  $E_T$  (denoted hereafter by  $d_T$ ) may increase and we will usually require an exponential growth in the number of particles  $P$  in order to obtain meaningful results, see e.g. Bickel *et al.* (2008). For instance, without the resampling step the ESS at time  $t+1$  is closely related to the following quantity (see e.g. Agapiou *et al.* (2015))

$$\rho_{t+1}(d_T) := \mathbb{E} \left[ \left( \prod_{s=1}^{t+1} \frac{\varphi_{s \rightarrow T}(\vartheta_{s-1 \rightarrow T})}{\varphi_{s-1 \rightarrow T}(\vartheta_{s-1 \rightarrow T})} \right)^2 \right],$$

which serves as a measure of the dissimilarity between proposals and targets, and that quite often increases exponentially in  $d_T$ . This quantity provides information about the limiting proportion of effective number of particles

since

$$\begin{aligned}
\lim_{P \rightarrow \infty} \left( \frac{\text{ESS}_{t+1}^P}{P} \right)^{-1} &= \lim_{P \rightarrow \infty} P \sum_{p=1}^P \left( w_{t+1 \rightarrow T}^{(p)} \right)^2 = \lim_{P \rightarrow \infty} \frac{\frac{1}{P} \sum_{p=1}^P \left( w_0^{(p)} \prod_{s=1}^{t+1} \frac{\varphi_{s \rightarrow T}(\vartheta_{s-1 \rightarrow T}^{(p)})}{\varphi_{s-1 \rightarrow T}(\vartheta_{s-1 \rightarrow T}^{(p)})} \right)^2}{\left( \frac{1}{P} \sum_{p=1}^P w_0^{(p)} \prod_{s=1}^{t+1} \frac{\varphi_{s \rightarrow T}(\vartheta_{s-1 \rightarrow T}^{(p)})}{\varphi_{s-1 \rightarrow T}(\vartheta_{s-1 \rightarrow T}^{(p)})} \right)^2} \\
&= \frac{\mathbb{E} \left[ \left( \prod_{s=1}^{t+1} \frac{\varphi_{s \rightarrow T}(\vartheta_{s-1 \rightarrow T})}{\varphi_{s-1 \rightarrow T}(\vartheta_{s-1 \rightarrow T})} \right)^2 \right]}{\left( \mathbb{E} \left[ \prod_{s=1}^{t+1} \frac{\varphi_{s \rightarrow T}(\vartheta_{s-1 \rightarrow T})}{\varphi_{s-1 \rightarrow T}(\vartheta_{s-1 \rightarrow T})} \right] \right)^2} = \rho_{t+1}.
\end{aligned}$$

The above equation implies that  $P = \mathcal{O}(\rho_{t+1}(d_T))$  if we want to maintain an acceptable level for the ESS. In our context, even though the targets  $\{\bar{\varphi}_{s \rightarrow T}\}_s$  are  $d_T$ -dimensional the ratios  $\{\varphi_{s \rightarrow T}/\varphi_{s-1 \rightarrow T}\}_s$  will involve cancellations of “fill in” variables as discussed in Section 2.1.2. This potentially leads to a much lower effective dimension of the problem than  $d_T$ , as in the case where targets follow (5).

For the SMC method presented in Dinh *et al.* (2016) in the context of phylogenetic trees, the authors have shown that  $\rho_T$  grows at most linearly in  $T$  under some strong conditions, somewhat comparable to the strong mixing conditions required in Theorem 2.2. Imposing an extra condition on the average branch length of the tree,  $\rho_T$  can be bounded uniformly in  $T$ . However, their method performs MH moves after resampling for improving the diversity of the particles, which could result in a sub-optimal algorithm. In contrast, TSMC uses MH moves for bridging  $\bar{\varphi}_t$  and  $\bar{\varphi}_{t+1}$  via the sequence of intermediate distributions  $\{\bar{\varphi}_{t,k}\}_{k=1}^K$ . Heuristically, the introduction of these intermediate distributions together with sensible transformations  $\{G_{t \rightarrow t+1}\}$  should alleviate problems due to the dissimilarity of targets, thus providing control over  $\rho_T$ .

In this respect, the authors in Beskos *et al.* (2014) have analysed the stability of SMC samplers as the dimension of the state-space increases when the number of particles  $P$  is fixed. Their work provides justification, to some extent, for the use of intermediate distributions  $\{\bar{\varphi}_{t,k}\}_{k=1}^K$ . Under some assumptions, it has been shown that when the number of intermediate distributions  $K = \mathcal{O}(d_T)$ , and as  $d_T \rightarrow \infty$ , the effective sample size  $\text{ESS}_{t+1}^P$  is stable in the sense that it converges to a non-trivial random variable taking values in  $(1, P)$ . The total computational cost for bridging  $\bar{\varphi}_t$  and  $\bar{\varphi}_{t+1}$ , assuming a product form of  $d_T$  components, is  $\mathcal{O}(P d_T^2)$ . Using this reasoning, we suspect TSMC will work well in similar and more complex scenarios, e.g. when the targets do not follow a product form or when strong mixing assumptions do not hold. This idea is supported with the examples that follow in sections 3 and 4.

## 3 Sequential Bayesian inference under the coalescent

### 3.1 Introduction

#### 3.1.1 Background

Inferring trees that represent the ancestry of a population is an important topic in phylogenetics and population genetics (Felsenstein, 2016). A Bayesian formulation of this problem is common (e.g. Drummond and Rambaut (2007)), with MCMC being used for inference. However, there is an important practical problem in this approach, in that it is not uncommon that one may analyse a dataset, then wish to update the results on receipt of new data (specific examples of this are below). Further, for some models exploring the space of possible trees is difficult, but it is the case that a tree that has high probability when using a subset of a dataset is informative about trees that have high probability when using the full data. To address either of these problems, we propose to use TSMC where a sequence of targets is constructed by adding the data sequentially, corresponding to adding leaves to a tree.

We anticipate this approach being useful in a range of models in phylogenetics and population genetics, but in this section we focus in particular on the problem of inferring the clonal ancestry of bacteria from their DNA sequences. With the recent wide availability of whole genome sequence data, this is a problem of great practical importance, with this approach being used when: studying the diversity of a whole species (e.g. Everitt *et al.* (2014)); studying small differences between closely related samples (e.g. Young *et al.* (2012)); or when using genetic differences to aid tracking the transmission of pathogens (e.g. Didelot *et al.* (2014)). In this latter case, one may wish to perform this tracking in real time, in order to make decisions that might stop the spread of an outbreak. Recent developments in sequencing technology mean that it is now possible to produce sequence data in a matter of hours, making the inference procedure the rate determining step. Even outside of this setting, due to the ease with

which data may be produced, it is very common for those working in genomics to wish to update their analysis as more data becomes available.

### 3.1.2 Previous work

The idea of updating a tree by adding leaves dates back to at least Felsenstein (1981), in which he describes, for maximum likelihood estimation, that an effective search strategy in tree space is to add species one by one. More recent work also makes use of the idea of adding sequences one at a time: ARGWeaver (Rasmussen *et al.*, 2014) uses this approach to initialise MCMC on (in this case, a space of graphs),  $t + 1$  sequences using the output of MCMC on  $t$  sequences, and TreeMix (Pickrell and Pritchard, 2012) uses a similar idea in a greedy algorithm. In work conducted simultaneously to our own, Dinh *et al.* (2016) also propose a sequential Monte Carlo approach in which the sequence of distributions is given by introducing sequences one by one. There are several differences to our approach, most of which are discussed in section 2.

Our SMC approach also yields an estimate of the marginal likelihood for each target distribution in the sequence. We note that SMC has something in common with the “stepping stones” approach Xie *et al.* (2011) used in population genetics, which is similar to AIS (i.e. an SMC sampler with MCMC moves and without resampling), where more than one MCMC move is used per target distribution.

### 3.1.3 Data and model

We consider the analysis of  $T$  aligned genome sequences  $y = y_{1:T}$ , each from a different individual. Each sequence consists of a string of characters with each character, taking a value in  $\{A, G, C, T\}$ , representing a base and being referred to as a “site”. The sequences being aligned results in them all being of equal length  $N$ , and in each character  $i$  (for  $1 \leq i \leq N$ ) representing the same base across all sequences. Many sites will be the same across all sequences, but some will differ across sequences. These sites are responsible for the genetic difference between the individuals, and they are referred to as single nucleotide polymorphisms (SNPs). The data used in our examples consists of seven “multi-locus sequence type” (MLST) genes of 23 *Staphylococcus aureus* sequences, which have been chosen to provide a sample representing the worldwide diversity of this species (Everitt *et al.*, 2014).

We make the assumption that the population has had a constant size over time, that it evolves clonally and that SNPs are the result of mutation. Our task is to infer the clonal ancestry of the individuals in the study, i.e. the tree describing how the individuals in the sample evolved from their common ancestors, and (additional to Dinh *et al.* (2016)) the rate of mutation in the population. We describe a TSMC algorithm for addressing this problem in section 3.2, before presenting results in section ?? . In the remainder of this section we introduce a little notation.

Let  $\mathcal{T}_t$  represent the clonal ancestry of  $t$  individuals and let  $\theta/2$  be the expected number of mutations in a generation. We are interested in the sequence of distributions

$$\pi_t(\mathcal{T}_t, \theta \mid y_{1:t}) \propto f(y_{1:t} \mid \mathcal{T}_t, \theta) p(\mathcal{T}_t) p(\theta),$$

where we use the coalescent prior (Kingman, 1982)  $p(\mathcal{T}_t)$  for the ancestral tree, the Jukes-Cantor substitution model (Jukes and Cantor, 1969) for  $f(y_{1:t} \mid \mathcal{T}_t, \theta)$  and choose  $p(\theta)$  to be a gamma distribution that has its mass on biologically plausible values of  $\theta$ . Appendix A contains full details of these choices, which are very standard in population genetics. Let  $l_t^{(a)}$  denote the length of time for which  $a$  branches exist in the tree, for  $2 \leq a \leq t$ . The heights of the coalescent events are given by  $h^{(a)} = \sum_{i=a}^t l_t^{(i)}$ , with  $h_t^{(a)}$  being the  $(t - a + 1)$ th coalescence time when indexing from the bottom of the tree. We let  $\mathcal{T}_t$  be a random vector  $(\mathcal{B}_t, h_t^{(2)}, \dots, h_t^{(t)})$  where  $\mathcal{B}_t$  is itself a vector of discrete variables representing the branching order. When we refer to a lineage of a leaf node, this refers to the sequence of branches from this leaf node to the root of the tree.

## 3.2 TSMC for the coalescent

In this section we describe an approach to adding a new leaf to an existing tree, described a transformation as in section 2.3. The basic idea is to first propose a lineage to add the new branch to, followed by a height  $h_t^{(\text{new})}$  conditional on this lineage at which the branch connected to the new leaf will join the tree. We describe particular designs for the distributions on these additional variables that lead to efficient algorithms (we note that Dinh *et al.* (2016) describe abstractly the types of different choices one might use in the phylogenetic case).

### 3.2.1 Transformation and weight update

Let  $g_t \sim \chi_t^{(g)}(\cdot \mid \theta_t, \mathcal{T}_t, y_{1:t+1})$  and  $h_t^{(\text{new})} \sim \chi_t^{(h)}(\cdot \mid g_t, \theta_t, \mathcal{T}_t, y_{1:t+1})$ . The transformation  $G_{t \rightarrow t+1}$  leaves  $\theta$ , and  $g_s, h_s^{(\text{new})}$  for  $s > t$ , unchanged. It makes a new tree from  $(\mathcal{T}_t, g_t, h_t^{(\text{new})})$  as follows. Firstly,  $g_t$  chooses a lineage to add the new branch to, where each possible lineage is indexed by the leaf on that lineage. Next we examine the coalescent heights. If  $\iota$  is such that  $h_t^{(\iota+1)} < h_t^{(\text{new})} < h_t^{(\iota)}$  then the effect of the transformation on the coalescence heights is

$$\left( (h_t^{(1)}, \dots, h_t^{(2)}) , h_t^{(\text{new})} \right) \mapsto \left( h_t^{(1)}, \dots, h_t^{(\iota+1)}, h_t^{(\text{new})}, h_t^{(\iota)}, \dots, h_t^{(2)} \right)$$

(or adding the new height to the beginning or the end of the vector if it is the first or last coalescence event), giving a Jacobian of 1. Then the new branching order is given by the original branching order, with a split in the branch that is uniquely determined by  $(\iota, g_t)$ , where the new branching order variable is denoted  $b_t^{(\text{new})}$ . We note that this transformation is not bijective: since branches higher up the tree are shared by multiple lineages there are multiple possible lineages that could have led to each tree.

Without loss of generality we examine the case of no intermediate distributions. In this case our weight update takes the form of equation 11, where we need to determine the ratio of  $\varphi_{t+1}$  and  $\varphi_{t \rightarrow t+1}$  evaluated at  $\vartheta_{t \rightarrow t+1}^{(p)}$ . As noted earlier in the paper,  $g_s, h_s^{(\text{new})}$  for  $s > t$  are not involved in the update. The variables involved are  $\mathcal{T}_{t+1}, \theta$ , which have resulted from the application of  $G_{t \rightarrow t+1}$ . To find  $\varphi_{t \rightarrow t+1}$  we must find the distribution under  $\varphi_t$  of the inverse image of  $\mathcal{T}_{t+1}, \theta$ . The resultant weight update is

$$\tilde{w}_{t+1} = w_t \frac{\pi_{t+1}(\mathcal{T}_{t+1}, \theta \mid y_{1:t+1})}{\pi_t(\mathcal{T}_t, \theta \mid y_{1:t}) \left[ \sum_{s \in \Lambda} \chi_t^{(g)}(g_t = s \mid \theta_t, \mathcal{T}_t, y_{1:t+1}) \chi_t^{(h)}(h_t^{(\text{new})} \mid g_t = s, \theta_t, \mathcal{T}_t, y_{1:t+1}) \right]}, \quad (20)$$

where  $\Lambda$  is the set that contains the leaves of the lineages that could have resulted in  $b_t^{(\text{new})}$ .

### 3.2.2 Design of auxiliary distributions

For our SMC sampler to be efficient, we must design  $\chi_t^{(g)}$  and  $\chi_t^{(h)}$  such that the distributions in the numerator and denominator of equation 20 are close, i.e. resulting in many trees that have high probability under the posterior with  $t+1$  sequences, but with the denominator having heavier tails than the numerator. In this method we first simulate the lineage then the height.

To choose the lineage, we make use of an approximation to the probability that the new sequence is  $M_s$  mutations from each of the existing leaves. Following Stephens and Donnelly (2000); Li and Stephens (2003) we choose the probability of choosing the lineage with leaf  $s$  using

$$\chi_t^{(g)}(s \mid \theta_t, y_{1:t+1}) \propto \left( \frac{N\theta_t}{t + N\theta_t} \right)^{M_s}. \quad (21)$$

This probability results from using a geometric distribution on the number of SNP differences between the new sequence and sequence  $s$  for each  $s$ , which is a generalisation of Ewens' sampling formula (Ewens, 1972) to the finite allele case. The geometric distribution results from integrating over possible coalescence times of the new sequence, yielding a choice for  $\chi_t^{(g)}$  that is likely to give our importance sampling proposal a larger variance than our target (a characteristic we observe in figure 1).

For  $\chi_t^{(h)}$  we propose to approximate the pairwise likelihood  $f_{t+1,s}(y_s, y_{t+1} \mid \theta, h_t^{(\text{new})}, g_t = s)$ , where  $y_s$  is the sequence at the leaf of the chosen lineage. Our likelihood is

$$L(h_t^{(\text{new})} \mid y_s, y_{t+1}, \theta_t) = \left[ \frac{3}{4} - \frac{3}{4} \exp(-2\theta_t h_t^{(\text{new})}/3) \right]^{M_s} \left[ \frac{1}{4} + \frac{3}{4} \exp(-2\theta_t h_t^{(\text{new})}/3) \right]^{N-M_s},$$

where  $M_s$  is the number of pairwise SNP differences between the new sequence and sequence  $s$ , both of length  $N$ . This likelihood may be approximated by a distribution using the Laplace approximation  $\mathcal{N}(\mu = \hat{h}, \sigma^2 = (-\hat{H})^{-1})$ , where  $\hat{h}_t^{(\text{new})}$  denotes the maximum likelihood estimate of  $h_t^{(\text{new})}$  and  $\hat{H}$  an estimate of the Hessian of the log likelihood at this estimate (Bishop, 2006). These estimates are given by

$$\hat{h} = -\frac{3}{2\theta_t} \log \left( 1 - \frac{4}{3} \hat{p} \right),$$

where  $\hat{p} = M_s/N$  is the proportion of sites that are different between the sequences, and

$$\hat{H} = \left( -\frac{M_s}{\hat{p}^2} - \frac{N - M_s}{(1 - \hat{p})^2} \right) \frac{2 \exp \left( -4\theta_t \hat{h}/3 \right)}{\theta_t}.$$

Reis and Yang (2011) proposes a more accurate approximation of the two sequence likelihood by using a Laplace approximation in a transformed space, in particular they propose  $2 \arcsin \sqrt{\frac{3}{4} - \frac{3}{4} \exp \left( -2\theta_t h_t^{(\text{new})}/3 \right)}$ . In this case the mean and variance of the Gaussian approximation are respectively  $\mu = \tilde{h}$  and  $\sigma^2 = \left( -\tilde{H} \right)^{-1}$  where

$$\tilde{h} = 2 \arcsin \sqrt{\frac{3}{4} - \frac{3}{4} \exp \left( -2\theta_t \hat{h}/3 \right)}$$

and

$$\tilde{H} = \frac{4\hat{H}}{\theta_t^2} \left( \frac{\cos \left( \frac{\tilde{h}}{2} \right) \sin \left( \frac{\tilde{h}}{2} \right)}{1 - \frac{4}{3} \sin^2 \left( \frac{\tilde{h}}{2} \right)} \right)^2,$$

modifying the derivation in Reis and Yang (2011). Thus to simulate  $h^{(\text{new})}$ , we first simulate  $\beta \sim \mathcal{N}(\cdot | \mu, \sigma^2)$ , then use  $h_t^{(\text{new})} = -\frac{3}{2\theta_t} \log \left( 1 - \frac{4}{3} \sin^2(\beta/2) \right)$ . The density of this distribution is given by

$$\begin{aligned} \chi_t^{(h)} \left( h_t^{(\text{new})} \right) &= \frac{1}{\sigma \sqrt{2\pi}} \exp \left( -\frac{\left( 2 \arcsin \sqrt{\frac{3}{4} - \frac{3}{4} \exp \left( -2\theta_t h_t^{(\text{new})}/3 \right)} - \mu \right)^2}{2\sigma^2} \right) \\ &\times \left( \frac{2\theta_t}{\sqrt{3} \sqrt{\exp \left( \frac{\theta h_t^{(\text{new})}}{3} \right)} - 1 \sqrt{\exp \left( \frac{\theta h_t^{(\text{new})}}{3} \right)} + 1 \sqrt{\exp \left( \frac{\theta h_t^{(\text{new})}}{3} \right)} + 3} \right). \end{aligned}$$

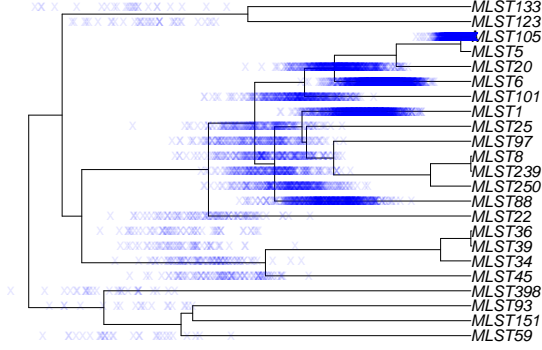
Figure 1 gives two examples of simulating new branch positions via the chosen  $\chi_t^{(g)}$  and  $\chi_t^{(h)}$ . In these examples we used the `coalescentMCMC` package in R to produce an empirical maximum likelihood estimate of the tree on our 23 *S. aureus* sequences. We then chose two examples in which we imagine that we have a tree estimated from 22 of the sequences, and wish to add the 23rd through simulating from our proposal distributions. Figure 1a shows 10,000 proposed positions for the new branch when adding the sequence with MLST type 5, and figure 1b 10,000 proposed positions when adding the MLST type 133 sequence. We found that we obtained a high proportion of probability mass in regions that have high mass in the posterior. For example, when proposing MLST 5, 46% of the proposal mass was on the lineage with leaf MLST 105, and we observe from figure 1a that most of the heights proposed on this lineage are close to the time of coalescence in the estimated tree. However, we also observe in both figures that the proposal is not too concentrated on particular lineages or heights, and by comparing figure 1a with 1b that the uncertainty is more pronounced when the new sequence is more distant from any existing sequence.

## 4 Bayesian model comparison for mixtures of Gaussians

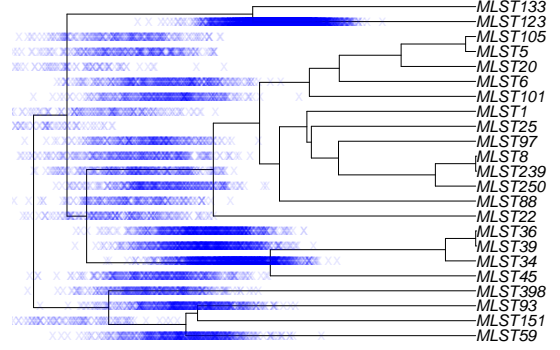
In this section we examine the mixture of Gaussians application in section 2.2: i.e. we wish perform Bayesian inference of the number of components  $t$ , and their parameters  $\theta_t$ , from data  $y$ . This application has been described thoroughly in a number of papers, including Richardson and Green (1997). Here we describe the application of TSMC to this problem, with the goal of illustrating the properties of TSMC. In order to easily highlight the properties of the new approach, we make two choices that do not necessarily lead to optimal performance in comparison to establish approaches to Bayesian inference for mixtures.

- Firstly, as mentioned previously we use the “without completion” model, in which we do not explicitly include a label  $z$  that allocates measurements  $y_i$  to a particular component. This has the effect of simplifying our parameter space, but increasing the complexity of our MCMC moves, when compared to the “with completion” model (which often exhibits superior performance (Jasra *et al.*, 2005a)).





(a) Draws of proposed branch positions for MLST 5.



(b) Draws of proposed branch positions for MLST 133.

Figure 1: Two examples of draws from the proposal distributions on lineage and height.

- Secondly, we do not address the lack of identifiability of the components and will observe that, as is desirable, TSMC explores the different modes in the parameter space due to this lack of identifiability.

In the next section (4.1) we outline the design of the algorithms used, then in section 4.2 we describe the results of using these approaches on the well-known galaxy data, highlighting features of the approach.

## 4.1 Description of algorithms

In this section we modify slightly the description of the model given in section 2.2. Our unknown parameters are the number of components  $k$ , and the parameters  $(\mu_{1:k}, \tau_{1:k}, \nu_{1:k})$  (means, precisions and weights respectively) of the  $k$  components for all  $k > 0$ , and our likelihood is the same as in equation 7. Here we do not assume that the weights  $\nu_{1:k}$  are known. We use priors

$$\begin{aligned}\tau &\sim \text{Gamma}(2, 1), \\ \nu_{1:k} &\sim \text{Dir}(1, \dots, 1)\end{aligned}$$

for the precisions and weights respectively, and for the means we choose an unconstrained prior of  $\mu \sim \mathcal{N}(m, S^2)$ , where  $m$  is the mean and  $S$  is the range of the observed data. We then impose an ordering constraint on the means, as described in Jasra *et al.* (2005b). This constraint is imposed solely for the purposes of improving the interpretability of our results. For simplicity we have also not included the commonly used “random beta” hierarchical prior structure on  $\tau$  (Richardson and Green, 1997). From a statistical perspective these choices are suboptimal: we emphasise that the only reason they are made here is to simply our presentation of the behaviour of TSMC.

We use different variants of TSMC (as described in section 2.3), using a sequence of distributions  $(\varphi_t)_{t=1}^T$  where  $\varphi_t(\vartheta_t) = \pi_t(\theta_t) \psi_t(u_t)$ .  $\pi_t$  is here the posterior on  $t$  components given by equation 8, and  $\psi_t$  is different depending on the transformation that is chosen.

Previous work has shown the advantages afforded by using intermediate distributions (as described in section 2.4.1), thus we use this approach (using geometric annealing) in all of our algorithms, making use of the adaptive method from section 2.4.3 to choose how to place these distributions. The results in this section focus particularly on illustrating the advantages afforded by making an intelligent choice of the transformation in TSMC. Sections 4.1.1 and 4.1.2 are devoted to describing, respectively, TSMC with the birth and split transformations. In section 4.1.3 we give details of the MCMC kernels used, which are the same when using both transformations.

### 4.1.1 Birth TSMC

First we consider the use of a birth move, which involves simply adding a component to the current set of components without altering the existing components (aside from renormalising the weights). Here  $u_t = (\mu_s, \tau_s, \nu_s)_{s=t+1}^T$  and

$\psi_t(u_t) = \prod_{s=t+1}^T p_\mu(\mu_s) p_\tau(\tau_s) b_s(\nu_s)$ , with  $b_s$  being a beta distribution  $\text{Beta}(1, s-1)$  (Richardson and Green, 1997). Then we let  $G_{t \rightarrow t+1}$  be the transformation that leaves all values of  $\mu$  and  $\tau$  the same, and leaves all  $\nu_s$  the same for  $s > t$ , but for  $1 \leq s \leq t$ , has  $\nu_s \mapsto \nu_s(1 - \nu_{t+1})$ . Then  $G_{t+1 \rightarrow t}$  leaves all values of  $\mu$  and  $\tau$  the same, and leaves all  $\nu_s$  the same for  $s > t$ , but for  $1 \leq s \leq t$ , has  $\nu_s \mapsto \nu_s / (1 - \nu_{t+1})$ .

Our reweighting step will be given by equation 17, which uses  $\varphi_{t \rightarrow t+1, k}$  as given in equation 14. This uses  $\varphi_{t \rightarrow t+1}$ , which is given by

$$\varphi_{t \rightarrow t+1}(\vartheta_{t \rightarrow t+1}) = \pi_t(\theta_{t+1 \rightarrow t}(\vartheta_{t \rightarrow t+1})) \psi_t(u_{t+1 \rightarrow t}(\vartheta_{t \rightarrow t+1})) |J_{t+1 \rightarrow t}|,$$

where  $\theta_{t+1 \rightarrow t}$  and  $u_{t+1 \rightarrow t}$  are the relevant parts of  $G_{t+1 \rightarrow t}$  above, and

$$J_{t+1 \rightarrow t} = (1 - \nu_{t+1})^{-(t-1)}.$$

#### 4.1.2 Split TSMC

Next we consider the use of a split move, in which a component is chosen to split into two, and where moment matching is used in order to intelligently replace the single component with two new components. We follow the approach in Richardson and Green (1997): let  $u_t = (u_{s,1}, u_{s,2}, u_{s,3})_{s=t+1}^T$  and  $\psi_t(u_t) = \prod_{s=t+1}^T b_1(u_{s,1}) b_2(u_{s,2}) b_3(u_{s,3})$ , with  $b_1$  and  $b_2$  both being the beta distribution  $\text{Beta}(2, 2)$  and  $b_3$  being the beta distribution  $\text{Beta}(1, 1)$ .

We now proceed by using the version of TSMC that allows the simultaneous use of multiple transformations; we will specify one transformation for each possible component split. We let  $\rho_t$  be the uniform distribution over the values  $(1, \dots, t)$ , and draw  $r_t \sim \rho_t$ . Then we let  $G_{t \rightarrow t+1}^{(r_t)}$  be the transformation that splits component  $r_t$ , the exact procedure for this being given in Richardson and Green (1997).

We now proceed as for the birth move, except that our weight update now depends on which component  $r_t$  we are splitting (we simply add the superscript  $r_t$  everywhere). Our reweighting step is given by equation 18, with  $\varphi_{t \rightarrow t+1, k}^{(r_t)}$  as given in equation 14. This uses  $\varphi_{t \rightarrow t+1}^{(r_t)}$ , which is given by

$$\varphi_{t \rightarrow t+1}^{(r_t)}(\vartheta_{t \rightarrow t+1}) = \pi_t\left(\theta_{t+1 \rightarrow t}^{(r_t)}(\vartheta_{t \rightarrow t+1})\right) \psi_t\left(u_{t+1 \rightarrow t}^{(r_t)}(\vartheta_{t \rightarrow t+1})\right) \left|J_{t+1 \rightarrow t}^{(r_t)}\right|,$$

where  $\theta_{t+1 \rightarrow t}^{(r_t)}$  and  $u_{t+1 \rightarrow t}^{(r_t)}$  are the relevant parts of  $G_{t+1 \rightarrow t}^{(r_t)}$  above, and  $J_{t+1 \rightarrow t}^{(r_t)}$  is given in Richardson and Green (1997).

#### 4.1.3 MCMC details

When moving from target  $t$  to  $t+1$  the initial step of the algorithm involves, for the birth move applying  $G_{t \rightarrow t+1}$  (or first simulating the component  $r_t$ , and applying  $G_{t \rightarrow t+1}^{(r_t)}$ ). Therefore our MCMC moves all act the space  $E_{t+1}$ . Recall that the variables  $u_{t \rightarrow t+1}$  will be updated by direct simulation from  $\psi_{t+1}$ , thus it remains to specify the updates on  $\theta_{t \rightarrow t+1}$ , which corresponds to the means, precisions and weights of  $t+1$  components. We use single-component Metropolis-Hastings moves on each  $\mu_s$  and  $\tau_s$ , and a Metropolis-Hastings move to update  $\nu_{1:t+1}$  jointly. For each  $\mu_s$  we use an additive normal random walk with proposal variance  $\sigma_{\mu_s}^2$  and for each  $\tau_s$  we use a multiplicative random walk, i.e. an additive normal random walk in log-space, with proposal variance  $\sigma_{\tau_s}^2$  in log-space. For  $\nu_{1:t+1}$  we follow Jasra (2005) and use an additive normal random walk in logit-space, with proposal variance  $\sigma_{\nu_s}^2$  in logit-space. We expect this final move to be efficient as long as  $t$  is not too large - to partially offset the decrease in efficiency as  $t$  increases, we use  $t$  iterations of this move at the  $(t+1)$ th iteration of the SMC.

In some experiments we use adaptive methods to estimate suitable proposal variances  $\sigma_{\mu_s}^2$ ,  $\sigma_{\tau_s}^2$  and  $\sigma_{\nu_s}^2$ , using sample variances from the previous SMC iteration. Let  $\hat{V}(\cdot)$  denote the function that takes the sample variance. At iteration  $(t+1, k+1)$  we use

$$\begin{aligned} \sigma_{\mu_s}^2 &= \hat{V}\left(\left\{\mu_s^{(p)}\right\}_{p=1}^P\right), \\ \sigma_{\tau_s}^2 &= \hat{V}\left(\left\{\log\left(\tau_s^{(p)}\right)\right\}_{p=1}^P\right), \\ \sigma_{\nu_s}^2 &= \hat{V}\left(\left\{\log\left[\frac{\nu_s^{(p)}}{1 - \sum_{l=1}^t \nu_l^{(p)}}\right]\right\}_{p=1}^P\right). \end{aligned}$$

## 4.2 Results

We ran four different TSMC algorithms on the enzyme data from Richardson and Green (1997): both birth and split TSMC, with and without adaptive MCMC proposals. We ran the algorithms up to a maximum of 5 components, with 1000 particles. We used an adaptive sequence of intermediate distributions, choosing the next intermediate distribution to be the one that yields a CESS (equation 19) of  $\beta P$ , where  $\beta = 0.9$ . We resampled using stratified resampling when the ESS (equation 2) falls below  $\alpha P$ , where  $\alpha = 0.5$ . When not using adaptive proposals, we chose  $\sigma_{\mu_s} = 0.1$ ,  $\sigma_{\tau_s} = 3$  and  $\sigma_{\nu_s} = 0.5$  for all  $s$  (these values being determined by pilot runs of the algorithms).

Figures 2 and 3 show results from these runs. Figure 2 illustrates in more detail the different behaviour of the birth and split algorithms (looking at the case where the MCMC moves are not adapted): it shows the evolution of the distribution of the component means when moving from one to two components. We observe that the split transformation has the effect of moving the parameters to initial values that are more appropriate for exploring the posterior on two components (note the different scales between the two cases), and the improvement in efficiency of the algorithm may be observed through the spacing of intermediate distributions used in both algorithms. This is performed automatically, ensuring that the distance between neighbouring distributions is not too large: in the birth case 53 intermediate distributions were used, compared to split, which used 23. We note that the split approach requires very few intermediate distributions after  $\gamma_k = 0.25$ , whereas the birth approach continues to require intermediate distributions right up to the end of the transition. Figures 2e and 2f provide further evidence of the improved properties of split TSMC, through showing the ESS over the SMC iterations. We note, however, that despite its poor initialisation, birth TSMC does provide an adequate representation of the posterior on two components, making use of intermediate distributions in order to avoid the sampler becoming degenerate.

Figure 3 shows the marginal likelihood estimates for each model (from the non-adaptive method, using birth moves). We note how useful TSMC may be for Bayesian model comparison in this setting. Suppose we wish to know the most probable model: we may run TSMC until we begin to see a decrease in the marginal likelihood (or posterior probability if we include a prior term) that we do not believe is due to Monte Carlo error in the estimates.

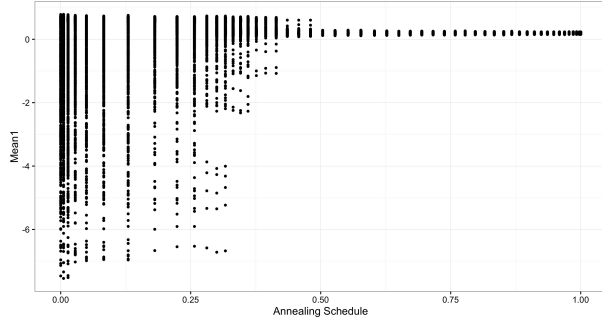
We now examine more closely the behaviour of the multiple routes used in split TSMC. We examine the case of moving from two to three components. At the beginning of the sequence of intermediate distributions, there is a 50-50 chance of a particle splitting the first or second component. There are 24 intermediate distributions used in this case, and after the 15th of these the particles become degenerate over the route variable. Recall that this does not affect the posterior, it simply reflects that the most effective choice of  $r_t$  becomes dominant as the algorithm progresses.

Finally we examine the case where the MCMC proposals are set adaptively by using particles from the preceding SMC iteration. Under the same conditions as the non-adaptive algorithms, the adaptive birth algorithm uses 54 intermediate distributions (compared to 53 in the non-adaptive case) and the adaptive split algorithm uses 20 intermediate distributions (compared to 23 in the non-adaptive case). Thus we see that the adaptive scheme is effective in the case of the split algorithm, where the distance between  $\varphi_t$  and  $\varphi_{t+1}$  is not too large. However, it doesn't improve over the arbitrarily chosen values when using the birth move, again due to the large distance between  $\varphi_t$  and  $\varphi_{t+1}$  in this case. For example, the posterior precision in the model with a single component will have a relatively low variance, leading to a small variance in the MCMC move on this parameter. However, when moving to two components, we require flexibility to explore the precision space widely, thus this is a poor proposal. Note that this property also makes  $\varphi_t$  a poor IS proposal for  $\varphi_{t+1}$ , with  $\varphi_t$  having lighter tails than  $\varphi_{t+1}$ . Thus, as noted above, the only means for the SMC to make an effective transition from  $\varphi_t$  to  $\varphi_{t+1}$  is to use MCMC moves in intermediate steps.

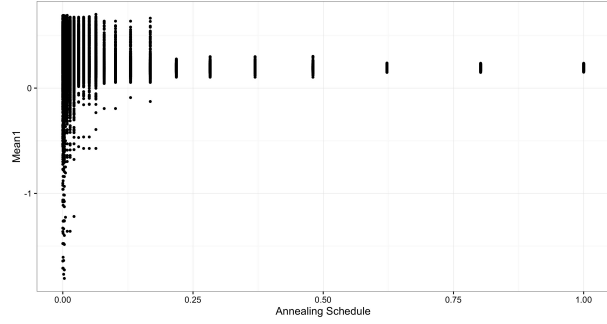
## 5 Conclusions

This paper introduces a new technique for Bayesian model comparison and parameter estimation, and an approach to online parameter and marginal likelihood estimation for the coalescent, underpinned by the same methodological development: TSMC. We show that whilst TSMC performs inferring on a sequence of posterior distributions with increasing dimension, it is a special case of the standard SMC sampler framework of Del Moral *et al.* (2007). In this section we outline several points that are not described elsewhere in the paper.

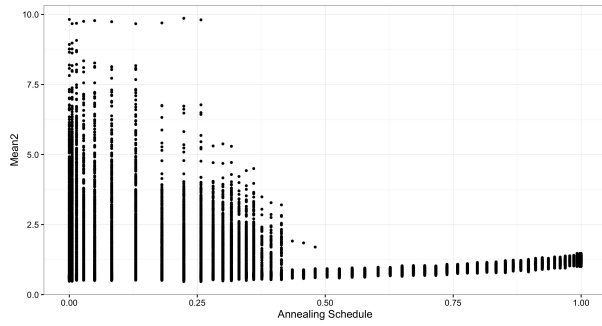
The innovation that distinguishes this variant of SMC samplers from most others described in the literature is in the use of transformations. In section 2.6 we see that the effectiveness of TSMC is governed by the distance between neighbouring distributions, thus to design TSMC algorithms suitable for any given application, we require the design of a suitable transformation that minimises the distance between neighbouring distributions. This is essentially the same challenge as is faced in designing effective RJMCMC algorithms, and we may make use of many of the methods



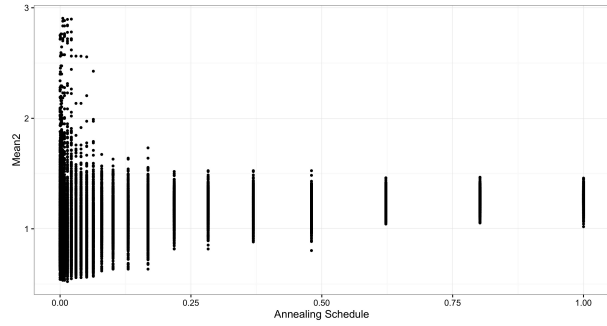
(a) The evolution of the estimated posterior on  $\mu_1$  (birth).



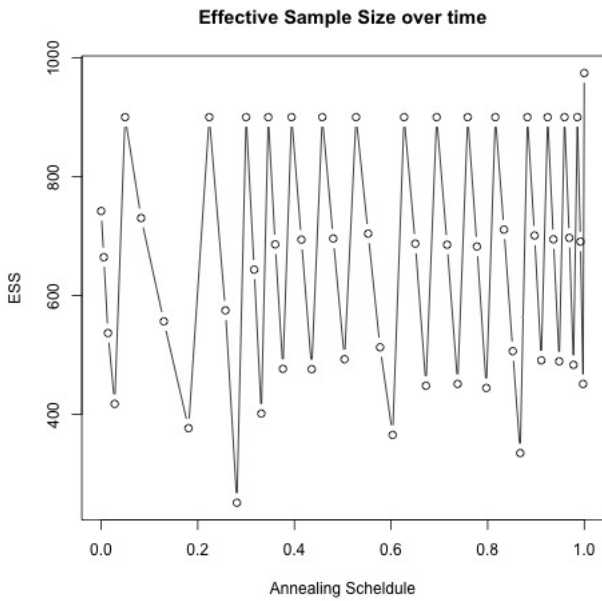
(b) The evolution of the estimated posterior on  $\mu_1$  (split).



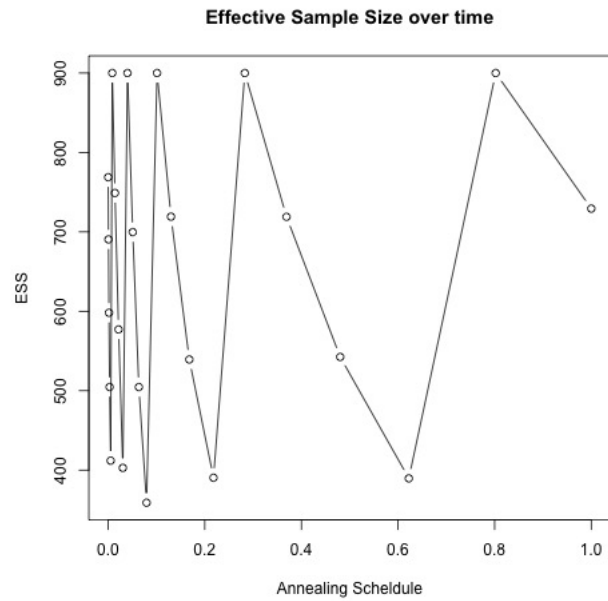
(c) The evolution of the estimated posterior on  $\mu_2$  (birth).



(d) The evolution of the estimated posterior on  $\mu_2$  (split).



(e) The evolution of the ESS (birth).



(f) The evolution of the ESS (split).

Figure 2: Comparing birth and split moves on the enzyme data.

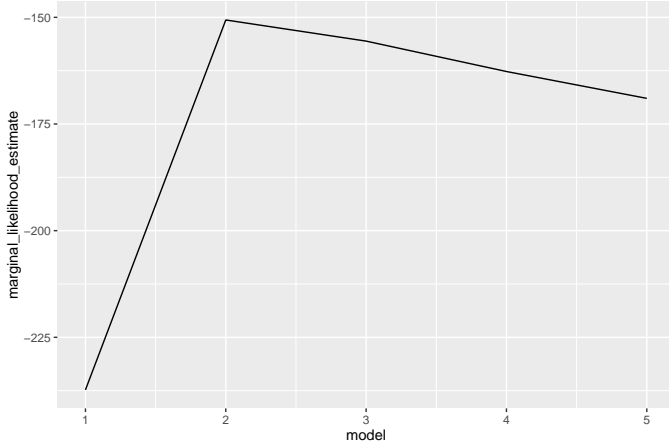


Figure 3: Marginal likelihood estimates from the enzyme data.

devised in the RJMCMC literature (Hastie and Green, 2012). The ideal case is to use a transformation such that every distribution  $\varphi_{t \rightarrow T}$  becomes identical, in which case one may simulate from  $\pi_T$  simply by simulating from  $\pi_0$  then applying the transformation. Approximating such a “transport map” for a sequence of continuous distributions is described in Heng *et al.* (2015).

One degree of flexibility not exploited in the methods described in the paper is that there is no formal requirement that we need to be able to simulate from the “fill in” distributions  $\psi_t$  for the variables  $u_t$ . It is sufficient to use Metropolis-Hastings updates on  $u_t$  for  $t > 0$ , although we do require to be able to simulate analytically from  $\psi_0$ .

In this paper we provide a simple application of our technique to population genetics, using the neutral coalescent model with mutation. It is well established in this field that when analysing many data sets, this model is inadequate. Many extensions exist, for example to account for the existence of recombination (where DNA is transferred horizontally between bacteria). These extended models often involve posterior distributions on spaces that have a much higher dimension than that considered here: the theoretical exploration in section 2.6 provides confidence that the use of our approach in these cases is not infeasible. Dinh *et al.* (2016) refer to the important issue of producing online alignments genome sequences: we do not consider this issue here, although it is an important practical issue in the case of the standard coalescent, and extensions. One other issue is how to decide the order in which to add sequences. This will undoubtedly affect the efficiency of the algorithm, but this effect will only be significant when the number of sequences is small (assuming that above this, the diversity existing set of sequences will, roughly speaking, encompass that of the sequences that are not yet seen). When the number of sequences is small (as in the case of a small number of mixture components), we observe that the algorithm will select many intermediate distributions to bridge the gap between  $\varphi_t$  and  $\varphi_{t+1}$ .

Although the examples in this paper both involve posterior distributions of increasing dimension, we also see a use for our approach in some cases that involve a distributions of decreasing dimension. For example, in population genetics, it is common to perform a large number of different analyses using different overlapping sets of sequences. For this reason many practitioners would value an inference technique that allows for the removal, as well as the addition, of sequences.

For Bayesian model comparison, we propose an approach that does not appear to have been considered previously, particularly in that path sampling estimates of the marginal likelihood may also be constructed from the SMC output (Zhou *et al.*, 2015). In figure 3 of section 4 we see a characteristic of this approach that will be common to many applications, in that the estimated marginal likelihood rises as the model is improved, then falls as the effect of the model complexity penalisation becomes more influential than improvements to the likelihood. We note that by using estimates of the variance of the marginal likelihood estimate (Lee and Whiteley, 2016), we may construct a formal diagnostic that decides to terminate the algorithm at a particular model, on observing that the estimated marginal likelihood declines from an estimated maximum value.

## Acknowledgements

Thanks to Christophe Andrieu, Adam Johansen and Changqiong Wang for useful discussions, and Xavier Didelot and Dan Lawson for establishing the novelty of the approach.

## A Likelihood and prior when using the coalescent with mutation

### A.1 Coalescent prior

The coalescent on  $t$  individuals uses a uniform distribution across branching orders and independent exponential distributions on the  $l_t^{(a)}$  (denoted by  $p_{l_t^{(a)}}$ ). Specifically, we have for  $2 \leq a \leq t$ ,

$$l_t^{(a)} \sim \text{Exp} \left( \frac{a(a-1)}{2} \right).$$

We use the alternative parameterisation  $h_t^{(a)}$ . Using the standard change of variables formula, the joint distribution  $p_{h_t}$  over  $h_t = (h_t^{(2)}, \dots, h_t^{(a)})$  is

$$p_{h_t}(h_t) = p_{l_t^{(t)}}(h_t^{(t)}) \prod_{a=2}^{t-1} p_{l_t^{(a)}}(h_t^{(a)} - h_t^{(a+1)}).$$

### A.2 Likelihood in the coalescent application

We define  $\theta/2$  to be the expected number of mutations in a generation (per site). To derive the likelihood used in this paper, we follow the argument on pages 446-447 of Felsenstein (2016), in which we consider a model for mutations occurring on the coalescent tree. To begin, we assume that our likelihood factorises over sites as

$$f(y_{1:t} \mid \mathcal{T}_t, \theta) = \prod_{i=1}^N f_i(y_{1:t,i} \mid \mathcal{T}_t, \theta),$$

where  $y_{1:t,i}$  is the sequence data at the  $i$ th site. We then need to specify the likelihood  $f_i$  at each site. This is given by introducing random variables describing distributions over “ancestral sequences”, these being the possible DNA sequences of the ancestors of the observed sample at each coalescence point.  $f_i$  is given by the joint distribution over  $y_{1:t,i}$  and the ancestral sequences at site  $i$  (conditional on  $\mathcal{T}_t, \theta$ ), marginalised over the ancestral sequences. The joint distribution is given by the product of a distribution over the sequence state at the root node of the tree (in this case we choose a uniform distribution over the four possible bases), and conditional distributions over the state of each ancestral sequence variable (or  $y_{1:t,i}$ ) given its parent in the tree. The marginalisation over ancestral sequence variables may be performed efficiently using Felsenstein’s pruning algorithm (Felsenstein, 1981), which is a special case of belief propagation (Pearl, 1988) applied to trees.

We spend the remainder of this section describing the conditional distributions of the sequence variables given their parents. Each of these takes the same form, depending on a parameter  $l$ , this being the length of the branch that connects each sequence variable to its parent. To derive the distribution, we look at the probability of an “event” happening: there are 4 possible events, corresponding to mutations from the current base to any of the 4 bases (one of these possibilities is that a mutation event happens, where the base “changes” from its current value to the same value). Instead of having a rate  $\theta/2$  of change to one of the three other bases, imagine instead that we have a rate  $(4/3) \times \theta/2 = 2\theta/3$  of change to a base randomly drawn from all four possibilities. This is exactly the same process: there is a probability of  $(\theta/2)/3$  of change to each of the other three bases (there is also an irrelevant rate  $(\theta/2)/3$  of change from a base to itself).

Letting the population size go to infinity, we obtain for a branch of length  $l$  in the genealogy, that the number of mutations on that branch is Poisson distributed with parameter  $2\theta l/3$ . From this Poisson distribution, we know that the probability of at least one event occurring is  $1 - \exp(-2\theta l/3)$ . The probability that it is any particular event that is a change is thus

$$\frac{1}{4} - \frac{1}{4} \exp \left( -\frac{2\theta}{3} l \right).$$

So, for each of the events when, for example, base  $A$  changes to a different base, we have the probability

$$P_{AX}(l) = \frac{1}{4} - \frac{1}{4} \exp \left( -\frac{2\theta}{3} l \right).$$

The probability of the base  $A$  staying the same can happen via two possibilities: there being no event, or via the event of changing base  $A$  to base  $A$ , thus we have the probability

$$\begin{aligned}
P_{AA}(l) &= \exp\left(-\frac{2\theta}{3}l\right) + \left(\frac{1}{4} - \frac{1}{4}\exp\left(-\frac{2\theta}{3}l\right)\right) \\
&= \frac{1}{4} + \frac{3}{4}\exp\left(-\frac{2\theta}{3}l\right).
\end{aligned}$$

## References

- Agapiou, S., Papaspiliopoulos, O., Sanz-Alonso, D., and Stuart, A. M. (2015). Importance Sampling: Computational Complexity and Intrinsic Dimension. *arXiv*.
- Alquier, P., Friel, N., Everitt, R. G., and Boland, A. (2016). Noisy Monte Carlo: Convergence of Markov chains with approximate transition kernels. *Statistics and Computing* 26(1), 29–47.
- Andrieu, C. and Roberts, G. O. (2009, apr). The pseudo-marginal approach for efficient Monte Carlo computations. *The Annals of Statistics* 37(2), 697–725.
- Andrieu, C. and Thoms, J. (2008). A tutorial on adaptive MCMC. *Statistics and Computing* 18(4), 343–373.
- Beaumont, M. a., Cornuet, J.-M., Marin, J.-M., and Robert, C. P. (2009, oct). Adaptive approximate Bayesian computation. *Biometrika* 96(4), 983–990.
- Beskos, A., Crisan, D., and Jasra, A. (2014). On the Stability of Sequential Monte Carlo Methods in High Dimensions. *The Annals of Applied Probability* 24(4), 1396–1445.
- Bickel, P., Li, B., and Bengtsson, T. (2008). Sharp failure rates for the bootstrap particle filter in high dimensions. In *Pushing the Limits of Contemporary Statistics: Contributions in Honor of Jayanta K. Ghosh*, Volume 3, pp. 318–329.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.
- Brooks, S. P. and Gelman, A. (1998). General Methods for Monitoring Convergence of Iterative Simulations. *Journal of Computational and Graphical Statistics* 7(4), 434–455.
- C  rou, F., Del Moral, P., and Guyader, A. (2011). A nonasymptotic theorem for unnormalized Feynman-Kac particle models. *Annales de l’Institut Henri Poincar  * 47(3), 629–649.
- Chopin, N. (2004, dec). Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *The Annals of Statistics* 32(6), 2385–2411.
- Del Moral, P. (2004). *Feynman-Kac Formulae*. Springer.
- Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society: Series B* 68(3), 411–436.
- Del Moral, P., Doucet, A., and Jasra, A. (2007). Sequential Monte Carlo for Bayesian Computation. *Bayesian Statistics*, 8, 1–34.
- Del Moral, P., Doucet, A., and Jasra, A. (2012). An adaptive sequential Monte Carlo method for approximate Bayesian computation. *Statistics and Computing* 22(5), 1009–1020.
- Didelot, X., Everitt, R. G., Johansen, A. M., and Lawson, D. J. (2011). Likelihood-free estimation of model evidence. *Bayesian Analysis* 6(1), 49–76.
- Didelot, X., Gardy, J., and Colijn, C. (2014). Bayesian inference of infectious disease transmission from whole genome sequence data. *Molecular Biology and Evolution*, 31, 1869–1879.
- Dinh, V., Darling, A. E., and Matsen IV, F. A. (2016). Online Bayesian phylogenetic inference: theoretical foundations via Sequential Monte Carlo. *arXiv*.
- Doucet, A. and Johansen, A. M. (2009). Particle Filtering and Smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 12, 656–704.

- Drummond, A. J. and Rambaut, A. (2007, jan). BEAST: Bayesian evolutionary analysis by sampling trees. *BMC Evolutionary Biology* 7(214).
- Everitt, R. G., Didelot, X., Batty, E. M., Miller, R. R., Knox, K., Young, B. C., Bowden, R., Auton, A., Votintseva, A., Larnier-Svensson, H., Charlesworth, J., Golubchik, T., Ip, C. L. C., Godwin, H., Fung, R., Peto, T. E. a., Walker, a. S., Crook, D. W., and Wilson, D. J. (2014, jan). Mobile elements drive recombination hotspots in the core genome of *Staphylococcus aureus*. *Nature Communications* 5(May).
- Everitt, R. G., Johansen, A. M., Roving, E., and Evdemon-Hogan, M. (2016). Bayesian model comparison with un-normalised likelihoods. *Statistics and Computing*.
- Ewens, W. J. (1972). The sampling theory of selectively neutral alleles. *Theoretical Population Biology*, **3**, 87–112.
- Fearnhead, P. and Taylor, B. M. (2013). An Adaptive Sequential Monte Carlo Sampler. *Bayesian Analysis* 8(1), 1–28.
- Felsenstein, J. (1981, jan). Evolutionary trees from DNA sequences: a maximum likelihood approach. *Journal of Molecular Evolution* 17(6), 368–376.
- Felsenstein, J. (2016). *Theoretical evolutionary genetics*. University of Washington.
- Flegal, J. M. and Jones, G. L. (2008). Batch means and spectral variance estimators in Markov chain Monte Carlo. (Mcmc), 1–41.
- Gelman, A. (1998). Some Class-Participation Demonstrations for Decision Theory and Bayesian Statistics. *The American Statistician* 52(2).
- Gordon, N. J., Salmond, D. J., and Smith, A. F. M. (1993). Novel approach to nonlinear/non-Gaussian Bayesian state estimation. In *Radar and Signal Processing, IEE Proceedings F*, Volume 140, pp. 107–113. IET.
- Green, P. (1995). Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. *Biometrika* 82(4), 711.
- Hastie, D. I. and Green, P. J. (2012). Model choice using reversible jump MCMC. *Statistica Neerlandica* 66(3), 309–338.
- Heng, J., Doucet, A., and Pokern, Y. (2015). Gibbs Flow for Approximate Transport with Applications to Bayesian Computation. *arXiv*, 1–30.
- Jasra, A. (2005). *Bayesian Inference for Mixture Models via Monte Carlo Computation*. Ph. D. thesis, Imperial College London.
- Jasra, A., Doucet, A., Stephens, D. A., and Holmes, C. C. (2008). Interacting sequential Monte Carlo samplers for trans-dimensional simulation. *Computational Statistics and Data Analysis* 52(4), 1765–1791.
- Jasra, a., Holmes, C. C., and Stephens, D. a. (2005a). Markov Chain Monte Carlo Methods and the Label Switching Problem in Bayesian Mixture Modeling. *Statistical Science* 20(1), 50–67.
- Jasra, A., Holmes, C. C., and Stephens, D. A. (2005b). Markov Chain Monte Carlo Methods and the Label Switching Problem in Bayesian Mixture Modelling. *Statistical Science* 20(1), 50–67.
- Jukes, T. H. and Cantor, C. R. (1969). *Evolution of Protein Molecules*. New York: Academic Press.
- Karagiannis, G. and Andrieu, C. (2013). Annealed Importance Sampling Reversible Jump MCMC Algorithms. *Journal of Computational and Graphical Statistics* 22(3), 623–648.
- Kingman, J. F. C. (1982). The coalescent. *Stochastic Processes and their Applications*, **13**, 235–248.
- Kong, A., Liu, J. S., and Wong, W. H. (1994). Sequential Imputations and Bayesian Missing Data Problems. *Journal of the American Statistical Association* 89(425), 278–288.
- Lee, A. and Whiteley, N. (2016). Variance estimation in the particle filter. *arXiv*.
- Li, N. and Stephens, M. (2003). Modeling Linkage Disequilibrium and Identifying Recombination Hotspots Using Single-Nucleotide Polymorphism Data. *Genetics*, **165**, 2213–2233.



- Neal, R. (2001). Annealed importance sampling. *Statistics and Computing* 11(2), 125–139.
- Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann.
- Pickrell, J. K. and Pritchard, J. K. (2012). Inference of Population Splits and Mixtures from Genome-Wide Allele Frequency Data. *PLoS Genetics* 8(11), e1002967.
- Raftery, A. E., Newton, M. A., Satagopan, J. M., and Krivitsky, P. N. (2006). Estimating the Integrated Likelihood via Posterior Simulation Using the Harmonic Mean Identity.
- Rasmussen, M. D., Hall, W., Hubisz, M. J., Gronau, I., and Siepel, A. (2014, may). Genome-wide inference of ancestral recombination graphs. *PLoS genetics* 10(5), e1004342.
- Reis, M. and Yang, Z. (2011). Approximate Likelihood Calculation on a Phylogeny for Bayesian Estimation of Divergence Times. *Molecular Biology and Evolution* 28(1969), 2161–2172.
- Richardson, S. and Green, P. J. (1997, nov). On Bayesian Analysis of Mixtures with an Unknown Number of Components (with discussion). *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 59(4), 731–792.
- Stephens, M. and Donnelly, P. (2000). Inference in molecular population genetics. *Journal of the Royal Statistical Society Series B* 62(4), 605–655.
- Xie, W., Lewis, P. O., Fan, Y., Kuo, L., and Chen, M.-H. (2011). Improving Marginal Likelihood Estimation for Bayesian Phylogenetic Model Selection. *Systems Biology* 60(2), 150–160.
- Young, B. C., Golubchik, T., Batty, E. M., Fung, R., Lerner-Svensson, H., Votintseva, A. A., Miller, R. R., Godwin, H., Knox, K., Everitt, R. G., Iqbal, Z., Rimmer, A. J., Cule, M., Ip, C. L. C., Didelot, X., Harding, R. M., Donnelly, P., Peto, T. E., Crook, D. W., Bowden, R., and Wilson, D. J. (2012). Evolutionary dynamics of *Staphylococcus aureus* during progression from carriage to disease. *Proceedings of the National Academy of Sciences* 109(12), 4550–4555.
- Zhou, Y., Johansen, A. M., and Aston, J. A. D. (2015). Towards automatic model comparison: an adaptive sequential Monte Carlo approach. *Journal of Computational and Graphical Statistics*.